



Review

Adaptive Learning with Gaussian Process Regression: A Comprehensive Review of Methods and Applications

Dominik Polke ^{1,*}, Elmar Ahle ¹ and Dirk Söffker ²

¹ Engineering and Computer Science, University of Applied Sciences Niederrhein, 47805 Krefeld, Germany; elmar.ahle@hs-niederrhein.de

² Chair of Dynamics and Control, University of Duisburg-Essen, 47057 Duisburg, Germany; soeffker@uni-due.de

* Correspondence: dominik.polke@stud.uni-due.de

Abstract

Gaussian processes (GPs) are a popular method in machine learning (ML) to model complex systems. One advantage of GPs over other ML models is their ability to quantify uncertainty in predictions. In the past, many advanced methods for GPs have been developed and published for various applications. Adaptive learning (ADL) is one of these applications, in which the consideration of uncertainty prediction plays a major role. The goal of ADL is to replace costly and time-consuming experiments and simulations of complex systems with surrogate models. This is achieved by strategically minimizing queries to maximize efficiency. In the ML literature, various reviews cover either GP methods or ADL strategies. Their focus is more on specific aspects. A comprehensive overview of different GP methods in various ADL applications was missing. This review categorizes GPs and related advanced methods for the first time in the context of ADL applications. A classification is provided for advanced GP methods, ADL methodologies, and practical application areas of GPs with ADL. This review distinguishes between ADL strategies with single-point and batch-query methods for Bayesian optimization and active learning, and highlights real-world applications such as material and product design, as well as efficient modeling for costly simulations and experiments. By combining these aspects, it offers a comprehensive guide for researchers and practitioners applying ADL with GPs to their specific use cases.

Keywords: machine learning; Gaussian process; adaptive learning



Academic Editor: Massimo Ferri

Received: 19 February 2026

Revised: 28 March 2026

Accepted: 30 March 2026

Published: 13 April 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

In complex physical processes and computationally demanding simulations, data acquisition is often challenging due to the need for extensive experimentation or time-intensive computations. To address this challenge, surrogate models are employed to provide accurate predictions, serving as efficient alternatives. Adaptive learning (ADL) offers a systematic approach to iteratively building models while solving optimization problems with high data efficiency, making it particularly valuable in scenarios where data collection is expensive. The primary objective of ADL is to minimize the number of queries needed for achieving the learning goal [1].

With Bayesian optimization (BO) [2] and active learning (ACL) [3], two different approaches to ADL [1] are known, each following distinct learning goals. The goal of BO is to efficiently optimize an unknown objective function by balancing exploration and

exploitation, while ACL focuses on improving model accuracy by selectively acquiring the most informative data points. Both compute surrogate models with efficient adaptive sampling schemes to accelerate the model building driven by a specific learning goal and undergo an exponential growth in popularity in the past decades [1].

The BO approach is a powerful tool for the optimization of design choices [4] and can be used for single- or multi-objective optimization goals in high-dimensional search spaces [5]. The application fields of BO range from product and material design optimization [6] to hyperparameter optimization of machine learning (ML) models [7] in an efficient way.

In contrast, ACL focuses on constructing accurate surrogate models. It enables sequential data acquisition by targeting regions of the parameter space with the highest information gain [8]. While traditionally adding one data point at a time [3], modern ACL techniques often query batches of data for improved efficiency in certain applications [9].

Both BO and ACL share an iterative approach, where a surrogate model is progressively refined, leveraging surrogate information to define a learning-goal-driven sampling process. This process is guided by an acquisition function that identifies the most promising regions of the input space for further exploration or exploitation [1].

A widely used choice for constructing surrogate models is the Gaussian process (GP), presented in [10]. The GP is effective for small data sets, balancing model complexity and generalization, while naturally handling uncertainties, making them suitable for ADL [11].

Standard GPs often rely on stationary covariance functions, which assume uniform smoothness across the input space. However, this assumption can lead to significant prediction errors in regions with abrupt variations, as shown by [12]. Non-stationary GPs address this issue by adapting to varying smoothness and heterogeneous data densities, as shown in [13], as well as [12]. These methods allow for improved modeling of complex input–output relationships, particularly in settings where the underlying function exhibits input-dependent smoothness or abrupt local variations, which are poorly captured by stationary assumptions [12,13].

Advanced GP methods, combined with ADL strategies, have been developed to further enhance surrogate modeling. For example, ADL techniques enable efficient exploration of high-uncertainty regions, improving prediction quality. Applications include sensitivity analysis, in which GPs are used to adaptively learn Sobol indices [14], and industrial scenarios, such as shape control of composite fuselages [15].

1.1. Reviews on Gaussian Processes

Gaussian processes have been extensively studied as a flexible Bayesian framework for regression, classification, and surrogate modeling. The comprehensive textbook [10] provides an introduction to GPs, covering fundamental concepts, practical implementations, and applications in ML.

In [16], a review of non-stationary GP surrogate models is presented, categorizing methods such as kernel adaptations, partition-based approaches, local GPs, and deep GP-based spatial warping. Additionally, publicly available software implementations are discussed, and a benchmark study is provided. In [17], a survey on deep Gaussian processes (DGPs) highlights their motivations, mathematical formulations, and advancements over the past decade. Key research directions and remaining challenges in DGP development are also discussed in [17].

Latent variable models based on GPs (GPLVMs) are reviewed in [18], where their connection to kernel principal component analysis is analyzed. The paper provides a taxonomy of GPLVM extensions and highlights their use in various ML applications. Challenges related to the scalability of GPs are discussed in [19], with a focus on global

approximation techniques such as sparse and structured kernel methods, as well as local approaches including mixtures of experts. This work also surveys recent advances aimed at enhancing the computational efficiency and scalability of GP models.

In [20], recent advances in GP regression methods are reviewed, particularly in handling large-scale and sparse data. The work highlights hierarchical low-rank approximations and Kronecker-structured GPs, with an example illustrating the performance of these methods with respect to accuracy and computational complexity. In [21], a survey on probabilistic surrogate modeling with GPs focuses on their role in risk assessment and computationally expensive simulations. The study reviews covariance parameter estimation techniques, validation criteria, and robust estimation methods.

A comprehensive overview and categorization of approaches for enforcing physical constraints in GP regression is provided in [22]. The survey covers methods for incorporating positivity or bound constraints, monotonicity and convexity constraints, as well as constraints based on linear partial differential equations (PDEs) and boundary conditions. In addition, the computational challenges associated with constrained GP models are extensively discussed, highlighting their importance for applications in scientific ML.

In [23], a focused review on GP-based model predictive control (GP-MPC) is presented. The use of GP regression for learning unknown dynamics in non-linear stochastic MPC is discussed. The literature is structured around three main challenges, namely scalable GP learning, uncertainty propagation over the prediction horizon, and closed-loop safety guarantees. In addition, GP-MPC pipeline designs and future research directions in dynamics learning and uncertainty quantification (UQ) are discussed.

These reviews collectively provide a structured perspective on the state of research in GPs, covering fundamental principles, advanced methods, scalability, constraints, and applications. This section summarizes their key contributions to establish the research landscape in GP-based modeling and surrogate learning.

1.2. Reviews on Adaptive Learning

Several surveys and reviews have been published on specific aspects of ADL, focusing on topics such as high-dimensional modeling, query strategies, and experimental design. In the survey [24], structural model assumptions for high-dimensional GP modeling in BO, including variable selection, additive decomposition, and low-dimensional embeddings are examined. In addition, modifications required for acquisition function optimization when handling high-dimensional spaces are discussed.

In [1], the synergy between BO and ACL is formalized as a unified adaptive learning framework. The work categorizes adaptive sampling techniques and demonstrates how Bayesian infill criteria and ACL criteria align as goal-driven methodologies across single and multi-fidelity learning scenarios. The survey in [25] provides a review of query strategies in ACL for classification, regression, and clustering within the pool-based ACL framework. Selection strategies are categorized into informative-based, representative-based, hybrid approaches, and more advanced techniques based on reinforcement learning and deep learning, while their mathematical foundations, applications, and challenges are analyzed.

The role of BO in experimental design is analyzed in [2], with a comparison to traditional Design of Experiments (DoE) methods. Key challenges are discussed, including the incorporation of prior knowledge, handling of constraints, batch evaluation, multi-objective optimization, and multi-fidelity data. In [26], approaches for scaling BO to high-dimensional spaces are examined, with a focus on methods that leverage structural properties of the objective function. Recent advancements that improve the efficiency of BO in scenarios with a high number of input dimensions are highlighted.

These reviews collectively provide a structured overview of ADL, covering both ACL and BO. They highlight advancements in handling high-dimensional spaces, integrating multiple fidelity levels, and aligning ADL with goal-driven optimization strategies. However, mainly specific aspects are addressed. They do not provide a comprehensive overview of GP-based ADL methods in regression-based surrogate modeling, which is the focus of the present review.

1.3. Scope of this Review

While GPs and ADL are widely studied in machine learning, existing reviews typically focus on specific aspects, such as theoretical foundations, scalable methods, or their role in BO and ACL, without providing a comprehensive overview of GP-based ADL. Surveys often analyze GPs independently or within BO and ACL but lack a systematic categorization of methods, sampling strategies, and applications. Additionally, previous reviews primarily address specialized topics, such as high-dimensional BO, domain-specific implementations, and specific ADL strategies. Consequently, a structured review of GP-based ADL methods, particularly in regression-based surrogate modeling, is lacking. This review addresses these gaps by systematically organizing and categorizing key aspects of GP-based ADL with a primary focus on regression tasks.

It provides a structured overview of advanced GP models, including scalable approximations, heteroscedastic, non-stationary, and multi-task GPs for improved surrogate modeling. Additionally, it examines ADL strategies, distinguishing between single-point and batch-query methods for BO and ACL, and highlights real-world applications such as hyperparameter optimization, material and product design, and efficient modeling for costly simulations and experiments. In addition, commonly used software libraries for implementing ADL with GPs are discussed. By combining these aspects, it offers a comprehensive guide for researchers and practitioners applying ADL with GPs to their use cases.

This review does not cover ML methods for ADL unrelated to GPs, such as neural networks or random forests, nor does it explore surrogate models that do not consider uncertainty-aware predictions. Additionally, general applications of ADL without the use of GPs are beyond the scope of this paper. The main contributions of this review are summarized as follows:

- **Advanced GP methods:** Categorizing advanced GP methods into non-stationary GPs, heteroscedastic GPs for variable noise estimation, scalable GP approximations, local GPs, multi-task GPs, dynamic GPs, and different training methods.
- **Learning strategies in ADL with GPs:** Categorizing adaptive learning strategies, including Bayesian optimization and active learning, and distinguishing between single-point and batch-query methods.
- **Applications of ADL with GPs:** Categorizing practical use cases, including hyperparameter tuning, material and product design, and efficient modeling for costly simulations and experiments.
- **Software libraries for GP-based ADL:** Surveying widely used Python and R packages for GPs, ACL, and BO, summarizing capabilities (batching, multi-output, heteroscedastic, and non-stationary modeling) and providing their key references.

This work is structured as follows. In Section 2, the fundamentals of GPs and their advanced methods are introduced. In Section 3, ADL and its integration with GPs are presented. In Section 4, the applications of ADL with GPs found in the literature are categorized. In Section 5, commonly used software libraries are discussed. The review concludes with a summary and an outlook on future research directions.

2. Gaussian Processes and Advanced Methods

In ML, probabilistic models that quantify uncertainty play a central role in robust decision making. Among these, GP regression (GPR) offers a flexible, non-parametric Bayesian framework that models distributions over functions instead of assuming a fixed functional form. In this section, the theoretical foundation of GP regression is introduced. In addition, recent developed advanced methods of GPs are presented.

2.1. Gaussian Process Regression

A GP is a non-parametric, probabilistic approach for modeling distributions over functions [10]. The GP generalizes the multi-variate normal to an infinite set of input points by defining a distribution over functions. A GP is defined as

$$f(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')), \quad (1)$$

where $m(\mathbf{x})$ denotes the mean function, $k(\mathbf{x}, \mathbf{x}')$ the covariance function, and $\mathbf{x} \in \mathbb{R}^{n_x}$ the input-point with n_x inputs. In practice, the kernel is typically specified by a finite set of hyperparameters θ (e.g., lengthscale, signal variance, noise variance).

In practice, the mean function is often assumed to be zero, $m(\mathbf{x}) = 0$, unless prior knowledge is used to choose a different structure. The covariance function $k(\mathbf{x}, \mathbf{x}')$ encodes assumptions about the function to be learned, such as smoothness, periodicity or stationarity. Common choices include the squared exponential (SE), the Matérn, and periodic covariance functions. For instance, the SE covariance function is given by

$$k(\mathbf{x}, \mathbf{x}') = \sigma_f^2 \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|^2}{2l^2}\right), \quad (2)$$

where $l \in \mathbb{R}^+$ denotes the lengthscale parameter and $\sigma_f^2 \in \mathbb{R}^+$ the signal variance. Both l and σ_f^2 are GP hyperparameters that are adapted to the data during training.

In real-world applications, observations are typically noisy. An additive Gaussian noise is assumed as

$$y_i = f(\mathbf{x}_i) + \epsilon_i, \quad \epsilon_i \sim \mathcal{N}(0, \sigma_n^2), \quad (3)$$

where $\sigma_n^2 \in \mathbb{R}^+$ denotes the noise variance.

With this noise model, the joint distribution of the noisy observations $\mathbf{y}$ and the latent function value at the test point results to

$$\begin{bmatrix} \mathbf{y} \\ f(\mathbf{x}^*) \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} m(\mathbf{X}) \\ m(\mathbf{x}^*) \end{bmatrix}, \begin{bmatrix} K(\mathbf{X}, \mathbf{X}) + \sigma_n^2 \mathbf{I} & k(\mathbf{X}, \mathbf{x}^*) \\ k^\top(\mathbf{X}, \mathbf{x}^*) & k(\mathbf{x}^*, \mathbf{x}^*) \end{bmatrix}\right), \quad (4)$$

where $f(\mathbf{x}^*)$ denotes the noise-free function value at the test input $\mathbf{x}^*$ and is treated as a random variable. The covariance matrix $K(\mathbf{X}, \mathbf{X}) \in \mathbb{R}^{N \times N}$ denotes the covariances between all pairs of training inputs. The vector $k(\mathbf{X}, \mathbf{x}) \in \mathbb{R}^{N \times 1}$ denotes the covariances between the training inputs and the test input point $\mathbf{x}$. The scalar $k(\mathbf{x}, \mathbf{x}) \in \mathbb{R}$ denotes the variance at the test input point $\mathbf{x}$. The vector $m(\mathbf{X}) \in \mathbb{R}^N$ denotes the mean function evaluated at all training inputs, and $m(\mathbf{x}) \in \mathbb{R}$ denotes the mean function evaluated at the test input point $\mathbf{x}^*$. The observation noise σ_n is added via the $N \times N$ identity matrix $\mathbf{I}$ on the variance.

Conditioning this joint distribution leads to the predictive posterior in the presence of noise

$$\mu(\mathbf{x}^*) = m(\mathbf{x}^*) + k^\top(\mathbf{X}, \mathbf{x}^*) [K(\mathbf{X}, \mathbf{X}) + \sigma_n^2 \mathbf{I}]^{-1} (\mathbf{y} - m(\mathbf{X})), \quad (5)$$

$$\sigma^2(\mathbf{x}^*) = k(\mathbf{x}^*, \mathbf{x}^*) - k^\top(\mathbf{X}, \mathbf{x}^*) [K(\mathbf{X}, \mathbf{X}) + \sigma_n^2 \mathbf{I}]^{-1} k(\mathbf{X}, \mathbf{x}^*). \quad (6)$$

A key advantage of GPs over many other regression methods is their ability to provide not only point predictions for unseen inputs x^* , but also a principled estimate of the associated uncertainty, both in the noise-free and noisy settings.

Beyond UQ, GPs are also increasingly required to respect physical or domain-specific constraints, such as equality, inequality, monotonicity, or boundary conditions, in order to avoid infeasible or physically inconsistent predictions. In [27], a recent probabilistic approach for constrained GP regression is presented, in which soft equality constraints are incorporated into the GP framework using Quasi Hamiltonian Monte Carlo (QHMC)-based sampling. The method combines probabilistic constraint handling with adaptive selection of informative constraint locations, and is positioned within a broader line of work in which related QHMC-based approaches have also been used for inequality and monotonicity constraints. This highlights probabilistic constraint enforcement as an active methodological direction in GP-based ML.

2.2. Advanced Methods of Gaussian Processes

In many real-world applications, the standard assumption of stationarity in GPs, i.e., constant statistical properties over the input space, does not hold. Data often exhibit non-uniform smoothness, heteroscedastic noise, temporal dependencies, or complex output relationships, which necessitate more flexible GP models. To address these challenges, various advanced methods of GPs have been developed, including non-stationary GPs (e.g., [28]), sparse GPs for scalability (e.g., [19]), dynamic GPs for dynamic system modeling (e.g., [29]), multi-output GPs for correlated outputs (e.g., [30]), local GPs for partitioned modeling (e.g., [13]), and GPs with approximated likelihoods such as Vecchia approximations (e.g., [31]). In this section, these methods are reviewed, categorized, and summarized with their modeling principles and limitations. A second perspective considered throughout this section is computational scalability. While exact GP inference provides high posterior fidelity, the cubic training cost becomes prohibitive on large data sets. The reviewed advanced GP variants therefore differ not only in modeling flexibility, but also in how computational bottlenecks are reduced and what may be sacrificed in terms of predictive accuracy, approximation quality, or additional modeling effort.

2.2.1. Anisotropic Gaussian Processes

In [10], GPs are introduced as a powerful non-parametric model using stationary covariance functions such as the squared exponential (SE) kernel. Many real-world machine learning applications involve a large number of input parameters, for example, in chemical engineering [32], autonomous driving [33], or energy consumption forecasting [34]. In such applications, the large number of input parameters results in a high-dimensional feature space, which poses challenges for distance-based learning algorithms due to the so-called curse of dimensionality [35]. An early enhancement to improve flexibility is automatic relevance determination (ARD) [36], where a separate lengthscale parameter is assigned to each input dimension, enabling anisotropic modeling and implicit feature selection [37].

The squared exponential covariance function with ARD is given by

$$k(x, x') = \sigma_f^2 \exp\left(-\frac{1}{2}(x - x')^\top L(x - x')\right), \quad (7)$$

where $L = \text{diag}([l_1 \dots l_{n_x}])^{-2}$ denotes the lengthscale matrix. Each $l_i \in \mathbb{R}^+$ determines the relevance of the corresponding input dimension.

Alternative feature selection approaches for GPs have also been proposed, as discussed in [38]. However, ARD remains a widely used and effective strategy, offering a principled way to perform feature relevance analysis within the probabilistic GP framework.

2.2.2. Non-Stationary Gaussian Processes

Various extensions have been proposed to address non-stationarity, differing in terms of flexibility, computational cost, and interpretability. Greater flexibility is often accompanied by increased computational cost. Input-dependent hyperparameters, nested latent structures, local partitions, or additional inference layers are frequently required in non-stationary models. As a result, improved representational power is often obtained at the expense of more demanding training and model selection. A direct extension to model non-stationarity is the use of input-dependent covariance functions. In [39], a non-stationary generalization of the Matérn covariance function is proposed, allowing for location-dependent correlation structures. A related approach is described in [40], where a two-level GP model is employed, with a second GP used to model input-dependent lengthscales of the primary GP. While effective, this method becomes computationally expensive in high-dimensional settings, as a separate second-level GP must be fitted for each input dimension.

Instead of modifying the kernel globally, other approaches divide the input space and train separate GPs locally. The Treed Gaussian Process (TGP) [41] partitions the space using a Bayesian decision tree and fits a GP in each region, an approach that has proven effective in ACL for computer experiments [42].

In [43], Gaussian Process Regression Networks (GPRNs) are introduced, combining Bayesian neural networks with the flexibility of GPs, allowing for input-dependent signal and noise correlations, lengthscales, and amplitudes across multiple outputs, providing a versatile framework for multivariate modeling.

The Deep GP (DGP) model introduced by [44] represents a Bayesian extension, stacking multiple GP layers in a compositional manner. Each layer transforms the input non-linearly, enabling hierarchical modeling of complex functions. Applications of ACL, such as in [45], demonstrate the modeling power of DGPs, although their training remains computationally demanding due to nested inference and scalability issues.

In [46], Deep Kernel Learning (DKL) is introduced, where the inputs of a stationary base kernel are transformed through a deep neural network, resulting in a highly expressive and scalable kernel representation.

A further development is proposed by [47], where a non-stationary generalization of the squared exponential covariance function is used. The covariance function used is defined as

$$k(\mathbf{x}, \mathbf{x}') = \sigma_f(\mathbf{x})\sigma_f(\mathbf{x}') \sqrt{\frac{2l(\mathbf{x})l(\mathbf{x}')}{l^2(\mathbf{x}) + l^2(\mathbf{x}')}} \exp\left(-\frac{|\mathbf{x} - \mathbf{x}'|^2}{l^2(\mathbf{x}) + l^2(\mathbf{x}')} \right), \quad (8)$$

where both the signal variance $\sigma_f(\mathbf{x}) \in \mathbb{R}^+$ and the lengthscale $l(\mathbf{x}) \in \mathbb{R}^+$ are input-dependent, enabling the model to flexibly capture local variations in the data. The input-dependent hyperparameters are learned in a full Bayesian manner with Hamiltonian Monte Carlo (HMC).

In [48], a non-stationary spectral kernel family is introduced by modeling the spectral density as a mixture of input-dependent GP frequency surfaces. The resulting generalized spectral mixture kernel captures non-stationary covariances and reduces to the stationary spectral mixture kernel when the frequency, lengthscale, and weight functions are constant.

Another class of methods uses deep neural networks (DNNs) to introduce non-stationarity into GPs. In [49], the Deep Gaussian Covariance Network (DGCN) is proposed, where a neural network predicts input-dependent GP hyperparameters. Although flexible, DNN-based models are less interpretable and pose challenges during training, as discussed in [50].

In [51], the Jump GP (JGP) models piecewise continuous functions by learning a local partition around each test location and using only same-region data for prediction. Partition and kernel hyperparameters are fitted by likelihood maximization via classification or variational expectation–maximization, which avoids smoothing across discontinuities.

A recent extension of JGP is given by Deep JGP (DJGP) [52], which is proposed for surrogate modeling of high-dimensional piecewise continuous functions. In DJGP, JGP is augmented by region-specific locally linear projection layers that map high-dimensional inputs to lower-dimensional local subspaces before the local JGP model is applied. In this way, DJGP addresses a key limitation of conventional JGP in higher-dimensional spaces, where piecewise continuous modeling becomes increasingly difficult due to the curse of dimensionality. By combining local dimensionality reduction with piecewise continuous GP modeling, DJGP extends the applicability of JGP to more challenging high-dimensional surrogate modeling tasks.

The Hierarchical Hyperplane Kernel GP (HHK-GP) proposed by [13] employs learnable hyperplane partitions to define areas for local GP experts. Local expert models are capable of modeling non-stationary behavior, although they introduce additional complexity through the joint optimization of multiple GP models and the hyperplane parameters. Despite this increased complexity, strong performance for ACL is reported in [13].

In [53], an Attentive Kernel GP (AKGP) makes a stationary base kernel non-stationary by mixing fixed length-scale base kernels with input-dependent attention and by suppressing correlations across discontinuities. The attention mappings and kernel parameters are learned by marginal likelihood optimization, which improves uncertainty calibration for information-gathering and mapping tasks.

A more interpretable solution is the Polynomial Chaos Expanded Gaussian Process (PCEGP) proposed by [28] and fully Bayesian by [54], where a polynomial chaos expansion (PCE) models the input-dependent lengthscale $l(\mathbf{x})$ and optionally the noise variance $\sigma_n^2(\mathbf{x})$. This results in an interpretable model of non-stationarity while maintaining an efficient runtime for training and prediction.

2.2.3. Sparse Gaussian Processes

The computational complexity of standard GPs scales cubically with the number of training points, which is described by $\mathcal{O}(N^3)$ due to the inversion of the covariance matrix [10]. This cubic scaling makes exact GP computationally expensive for large training sets [10]. In many real-world applications, especially those involving large data sets, this computational bottleneck becomes prohibitive. To overcome this limitation, sparse GPs have been developed that introduce a set of inducing points summarizing the information of the full data set, thereby reducing computational complexity while maintaining predictive accuracy [55]. Exact posterior inference is replaced by an inducing-point representation, which improves scalability substantially. Predictive fidelity, however, depends on how well the inducing set captures the relevant structure of the full data set. Computational efficiency is, therefore, gained at the cost of approximation error and additional design choices regarding the inducing-point representation.

The variational sparse GP (VSGP) framework, introduced by [55], formulates sparse inference as a variational optimization problem, allowing efficient training and posterior sampling via Markov-Chain-Monte-Carlo (MCMC). A theoretical comparison of common sparse schemes, such as the Fully Independent Training Conditional (FITC) and Variational Free Energy (VFE) approximations, is provided by [56], clarifying their respective trade-offs between computational efficiency and approximation quality.

Applications to specific ML tasks have also been developed using sparse GP methods. In the context of Bayesian optimization, a sparse online GP is proposed in [57], where

weighted inducing points are updated to focus computational resources on promising regions of the input space. For photometric redshift estimation under heteroscedastic noise, a GP approach called GPz is presented in [58], demonstrating how sparse approximations can be extended to handle input-dependent uncertainties.

Extensions to multi-output settings are addressed in MedGP [59], where sparse GPs are used for real-time prediction of medical time series by modeling multiple correlated outputs efficiently. Recursive sparse modeling is proposed by [60], where additive local GPs are combined hierarchically to capture complex structure across different scales. Applications in scientific domains further highlight the versatility of sparse GPs, such as their use in materials science for accelerating materials discovery [61] and in transfer learning scenarios, where sparse GPs enable efficient knowledge transfer between tasks [62].

Despite their advantages, sparse GPs introduce additional modeling choices regarding the number, placement, and optimization of inducing points. Improved scalability is thereby achieved in comparison with exact GPs, but part of the modeling burden is shifted to approximation design. Practical success is, therefore, determined by how well computational savings and approximation quality are balanced.

2.2.4. Dynamic Gaussian Processes

To model dynamic systems, GP models have been extended to incorporate time-varying structure. Such models are particularly relevant in scenarios where UQ is critical, for instance, in control systems, system identification, and emulation of physical processes. Standard GPs assume independent and identically distributed inputs, which limits their effectiveness for sequential or state-dependent tasks. Dynamic GP models address this by explicitly accounting for temporal correlations and latent state transitions.

A comprehensive overview of GP-based modeling and control of dynamic systems is presented in [29], introducing both non-linear autoregressive with exogenous input (NARX) models and state-space formulations based on GPs. These structures allow for flexible modeling of time-dependent behavior while maintaining a probabilistic treatment of uncertainty. In model-based control, GPs are integrated with model predictive control (MPC) in [63], enabling uncertainty-aware trajectory optimization. This approach leverages GP-based dynamics models to ensure safe and data-efficient decision-making.

Comparative studies of dynamic GP architectures are conducted in [64], where NARX, state-space, and MPC-driven GP models are evaluated in terms of predictive accuracy and control performance in simulation environments. In [65], the training process of state-space GP models is investigated and the impact of different training choices on the resulting closed-loop control performance is analyzed.

Further applications of dynamic GPs include digital twin frameworks, where GPs are used as real-time surrogates for physical systems. In [66], it is demonstrated how GPs can emulate dynamic processes robustly under measurement noise and system variability, supporting adaptive control and system monitoring.

2.2.5. Multi-Output Gaussian Processes

In many applications, multiple related quantities must be predicted simultaneously. Standard GP models treat each output independently, ignoring potential correlations. Multi-output GPs (MOGPs) extend the GP framework to jointly model vector-valued functions, capturing dependencies across outputs to improve predictive performance and UQ. This is particularly beneficial when outputs are physically or statistically coupled, such as manufacturing and product development. Increased predictive efficiency may be obtained through shared structure across outputs. At the same time, greater computational and modeling complexity was introduced in comparison with independent single-output

GPs. Trade-offs, therefore, arise in kernel design, inference cost, and scalability, especially when many outputs or heterogeneous likelihoods are involved.

The treed multi-output Gaussian process (MOGP) model by [67] simplifies cross-output dependencies by treating outputs as conditionally independent given a common covariance function, assuming similar output regularity, which reduces computational costs. The approach employs adaptive, non-probabilistic input space partitioning, training local multi-output surrogates in each region. This design improves scalability and flexibility, particularly in heterogeneous domains, enabling the model to capture non-stationary responses, locate discontinuities, and identify localized features

In the context of causal inference, a multi-task GP defined in a vector-valued reproducing kernel Hilbert space is used in [68] to estimate treatment effects, while factual and counterfactual outcomes are learned simultaneously. In [69], the proposed Multi-Output Spectral Mixture (MOSM) kernel extends existing approaches by modeling phase-shifted dependencies and delays across outputs through the direct design of cross-covariances as a spectral mixture kernel. This is achieved using a parametric family of complex-valued cross-spectral densities based on Cramér's Theorem, providing a clear interpretation of these delays and phase differences among outputs.

Methods for heterogeneous outputs with output-specific likelihoods and inducing-point-based approximations to enable scalable inference in high-dimensional multi-output settings are developed by [30]. An overview of key challenges in MOGP modeling, including kernel construction, scalability, and task heterogeneity, is given in [70].

In [71], adaptive sampling in multi-task learning GPs is addressed for manufacturing contexts, improving data efficiency by exploiting task similarities. In [59], the approach MedGP is proposed, a sparse multi-output GP implementation that explicitly models temporal dynamics and cross-variable correlations in clinical time series data.

2.2.6. Local Gaussian Processes

In large-scale or highly non-stationary problems, global GP models often become computationally intractable or fail to capture localized behaviors. Local GP methods address this by dividing the input space into smaller regions and fitting separate GP models within each region. This enables improved scalability, local adaptivity, and efficient parallelization, particularly for complex or spatially heterogeneous functions.

A classical approach is the Treed Gaussian process (TGP) model [41], which employs Bayesian decision trees to partition the input space and trains independent GPs in each leaf. This method has been successfully applied to ACL and surrogate modeling of computer experiments in [42].

Local approximation techniques have also been proposed for scalable inference. In [72], a local GP framework is described that dynamically selects relevant data subsets and supports parallel training for large data sets. A patching approach for massive spatial data is implemented in [73], where the domain is decomposed into spatially coherent patches, each modeled by a separate GP.

A local partitioning approach in [51] builds a boundary near each test input and selects only same-region neighbors for estimation. Boundary and GP hyperparameters are learned by likelihood maximization using classification or variational expectation-maximization, which reduces bias at jumps [51].

Domain-specific applications further illustrate the utility of local models. In [74], the local approximate Gaussian process regression (laGPR) method is applied to data-driven constitutive modeling in multiscale mechanics, and superior performance is demonstrated in comparison to artificial neural networks. In sensor-based applications with small-sample data, prediction accuracy is shown to be enhanced by local GPs in [75].

More recently, the Hierarchical Hyperplane Kernel GP (HHK-GP) is introduced in [13], in which partitions are defined via learned hyperplanes and local GP experts are jointly optimized. Both TGP and HHK-GP are non-stationary and based on local GPs.

Improved scalability and adaptivity are obtained in local GP models by restricting inference to smaller regions or neighborhoods. In return, global consistency is partially traded for local efficiency. Predictive quality is therefore strongly influenced by the construction of partitions or neighborhoods and by the degree to which boundary effects and coordination across regions can be controlled. The computational burden of exact global inference is reduced, but additional partitioning complexity and potentially reduced global coherence must be accepted.

2.2.7. Vecchia Approximation and Fast Inference

When the number of training data points n becomes large, standard GP inference turns computationally prohibitive due to the cubic scaling $\mathcal{O}(n^3)$, which arises from matrix inversions. To address this challenge, the Vecchia approximation introduces conditional independence assumptions to factorize the joint probability distribution, thereby enabling scalable inference. Originally developed in spatial statistics, this approach is extended in [76] and further generalized in [77], where a unifying framework for various structured GP approximations is presented.

In the context of Bayesian optimization, the Vecchia approximation is adapted in [78] to support mini-batch training and efficient neighbor selection, making it suitable for high-throughput scenarios.

To further accelerate prediction, the Lanczos Variance Estimates (LOVE) algorithm is proposed in [79] to approximate posterior variances efficiently using linear algebra techniques. Substantial reductions in computational cost are obtained by both methods through structural approximations or fast linear algebra shortcuts. The corresponding trade-off is that predictive quality depends more strongly on the suitability of the underlying approximation assumptions for the data. Losses in posterior fidelity may therefore occur when conditional independence assumptions, neighbor structure, or matrix approximation quality are not well aligned with the underlying problem.

2.2.8. Further Advanced Methods

Recent developments combine GPs with other ML techniques. In [80], PCE-Kriging is introduced as a hybrid surrogate modeling approach that combines PCE and Kriging. In this method, the global behavior of the computational model is represented by PCE using a sparse set of orthonormal polynomials tailored to the input distributions, while the local variability of the output is modeled by Kriging. The optimal sparse polynomial basis is selected through an adaptive algorithm similar to least angle regression. Numerical benchmarks demonstrate that PCE-Kriging outperforms or matches the accuracy of the standalone methods, especially when the number of training samples is limited, making it attractive for computationally expensive models.

In [81], GPBoost is proposed as a method that integrates gradient boosting with GPs and mixed effects models. This approach relaxes the common zero or linearity assumptions for the prior mean function in GP models and the independence assumption in boosting algorithms. The method improves prediction accuracy, effectively handles high-cardinality categorical variables, and yields probabilistic predictions. An extension based on the Vecchia approximation enables scalability to large data sets [81]. Experimental results show superior predictive performance across a range of simulated and real-world applications compared to existing methods.

2.2.9. Overview of Advanced Gaussian Process Models

In Table 1, the reviewed advanced GP models are summarized and categorized into standard, non-stationary, sparse, dynamic, multi-output, local, and fast approximate methods. For each method, key characteristics and limitations are listed, along with representative references, providing a structured overview of modeling principles and trade-offs. Within each category, the methods are ordered by the publication year.

Table 1. Overview of advanced Gaussian process models.

Category	Method	Characteristics	Limitations	References
Standard	GP with ARD	Anisotropic input weighting, implicit feature selection	Stationary kernel	[10]
Non-Stationary	Non-stationary kernel	Location-dependent correlations	Computational cost in high dimensions	[39,40]
Non-stationary	TGP	Bayesian tree partitions, local experts	Tree growth complexity	[41]
Non-Stationary	GPRN	Non-stationary correlations and noise, multi-output	Complex inference, tuning	[43]
Non-Stationary	Deep GP	Hierarchical layers, compositional learning	Nested inference, scalability	[44,45]
Non-Stationary	Deep kernel learning	Neural input transformation	DNN structure selection, low interpretability	[46]
Non-Stationary	Non-stationary and heteroscedastic GP	Non-stationary correlations and noise	Computational cost with HMC	[47]
Non-Stationary	Non-stationary spectral kernel	Input-dependent spectral density, non-stationary and non-monotonic covariances	Model complexity, initialization sensitivity	[48]
Non-Stationary	DGCN	DNN hyperparameter estimation	Low interpretability, DNN structure selection	[49]
Non-stationary	JGP	Local partitioning, handles discontinuities	Partition learning overhead	[51]
Non-Stationary	HHK-GP	Hyperplane-based local experts	Complex joint optimization	[13]
Non-Stationary	Attentive Kernel GP	Input-dependent attention mixes fixed-scale base kernel, masks cross-region correlations	Model complexity, choice of primitive scales and network tuning	[53]
Non-Stationary	PCEGP	PCE hyperparameter estimation, interpretable	PCE basis selection, scalability in high-dimensions	[28,54]
Non-Stationary	DJGP	Region-specific locally linear projection layers for high-dimensional piecewise continuous modeling	Increased model complexity and inference effort	[52]
Sparse	FITC/VFE	Approximation schemes, flexible	Bias, sensitive tuning	[56]
Sparse	GPz	Sparse model and heteroscedastic noise	Domain-specific tuning	[58]
Sparse	VSGP	Variational inference, scalable posterior sampling	Inducing point selection critical	[55]
Sparse	Online sparse BO	Adaptive inducing point updates	Sensitivity to initialization	[57]
Sparse	MedGP	Multi-output, temporal dynamics	Limited to time series	[59]
Sparse	Sparse Additive GP	Hierarchical, additive modeling	Partitioning choices	[60]
Sparse	Transfer sparse GP	Transfer learning, inducing point selection algorithm	Limited to homogenous domains	[62]

Table 1. Cont.

Category	Method	Characteristics	Limitations	References
Dynamic	State-space GP	dynamic system modeling, latent states	One GP for each state needed	[29,64,65]
Dynamic	NARX-GP	Non-linear autoregression, feedback	High model complexity	[29,64]
Dynamic	Cautious GP-MPC	Uncertainty aware MPC, sparse GP	Limited to one application	[63]
Dynamic	Digital twin GP	Real-time emulation, adaptive control	Sensitive to data quality	[66]
Multi-Output	Treed MOGP	Adaptive partitions, local surrogates	Hyperparameter optimization of tree	[67]
Multi-Output	Multi-task GP	Individualized measures of confidence, causal inference as a multi-task learning problem	Limited to medical use-case	[68]
Multi-Output	Spectral kernel MOGP	Parametric family of complex-valued cross-spectral densities	Spectral kernel tuning required	[69]
Multi-Output	Hetero MOGP	Output-specific likelihoods	Scalability limits	[30]
Multi-Output	Sparse MOGP	Multi-output, temporal and cross-variable structure	Output alignment required	[59]
Multi-Output	MOGP with adaptive sampling	Task allocation, data efficiency	Assumes task similarity	[71]
Local	TGP	Bayesian tree partitions, local experts	Tree growth complexity	[41]
Local	Local GP approx.	Region-wise fitting, parallelizable	Needs coordination across regions	[72]
Local	Patching GPs	Spatial patches, smooth boundaries	Edge inconsistency	[73]
Local	laGPR	Integration in non-linear finite element setting	Scalability with data set size	[74]
Local	HHK-GP	Learned hyperplane partitions	Joint optimization	[13]
Vecchia Approx.	Vecchia GP	Linear-time, spatial factorization	Global consistency loss	[76,77]
Vecchia Approx.	BO with Vecchia	Mini-batch, scalable BO	Approximation artifacts	[78]
Fast Inference	LOVE approx.	Fast variance estimation	Variance approximation error	[79]
Further Methods	PCE-Kriging	Global trend modeling with PCE, local variability captured by GP	High model complexity	[80]
Further Methods	GPBoost	Gradient boosting, mixed effects	Optimization of tree and boosting needed	[81]

3. Adaptive Learning

Data-efficient, goal-driven modeling is essential in scenarios where data acquisition is costly or limited. In this context, methods for sequentially selecting data points are employed either to improve a surrogate model through ACL or to optimize an objective function via BO. Recent work, particularly by [1], has unified these approaches within the framework of goal-oriented learning.

This section covers fundamental concepts, initial design strategies, and acquisition methods using GPs. The objectives and methodologies of ACL and BO are distinguished, and their conceptual overlap is emphasized.

3.1. Fundamentals and Definitions

Adaptive learning refers to a class of data-efficient, goal-driven methodologies that iteratively collect data to achieve a predefined objective. This approach plays a central role

in scientific and engineering applications, where objective functions such as performance metrics or error indicators are typically expensive to evaluate, either through costly physical experiments or through computationally intensive simulations. In such settings, ADL leverages surrogate models and informed sampling strategies to minimize the number of expensive evaluations required [3].

A foundational distinction within adaptive strategies can be made between one-shot designs and sequential designs. One-shot designs such as Latin hypercube, factorial, and Sobol designs select all sample points upfront. These approaches are often used when no feedback is available during the data acquisition phase. Sequential or adaptive sampling strategies are favored, as they allow data to be collected iteratively, refining the surrogate model and focusing sampling efforts where they are most useful [1].

Within this adaptive setting, two major methodologies have emerged, namely ACL and BO. Both aim to strategically select new input locations that are most informative for improving a model or achieving an optimization goal. The primary objective of ACL is to reduce uncertainty or enhance model predictions across the entire domain, making it particularly suited for scenarios where the goal is to improve predictive performance with minimal labeled data. In contrast, the focus of BO lies in identifying the global optimum of a black-box objective function. To achieve this, a surrogate model of the unknown function is constructed and employs an acquisition function to balance exploration and exploitation when selecting new data points.

In [1], this distinction is unified under the concept of goal-driven learning, where each sampling decision is made to acquire information that contributes most to a specific objective. From this perspective, both ACL and BO can be viewed as instances of a broader adaptive learning framework, differing primarily in how they define and pursue their goals.

Recognized as a powerful and general-purpose solution for design problems across scientific and industrial domains [4], BO has been successfully applied in fields such as hyperparameter optimization [82], robotics [83], drug discovery [84], environmental monitoring [85], sensor placement [86], materials science [87], and chemistry [88]. These applications frequently involve high-dimensional, constrained, and black-box objective functions that lack analytical gradients and are costly to evaluate. The strength of BO lies in its sample efficiency and its ability to balance exploration and exploitation in such uncertain environments [2].

In particular, BO has been shown to be effective in experimental design under constraints, mixed variable types, and multi-objective formulations, which are common in scientific domains where feasibility constraints arise from physics, chemistry, or domain-specific considerations [2]. In materials science, for example, optimal experimental design is subject to composition constraints to avoid undesired material properties. As discussed in [89], the exploration of high-dimensional chemical spaces can be guided by ACL under known constraints on component ratios.

Sampling strategies in ACL are often based on selecting new points from a candidate set, guided by the current state of the surrogate model. In classical settings, a finite pool of candidates is used, from which the most informative samples are selected according to an acquisition function. In contrast, many modern ADL approaches with GPs operate directly in continuous input spaces, where new points are determined by optimizing the acquisition function over the entire domain. While this no longer corresponds to a fixed pool in the strict sense, the methodology remains structurally similar and is often viewed as an extension of pool-based ACL. The process remains goal-driven, model-based, and iterative. As proposed in [1], the synergy between these methods enables a unified understanding and joint advancement of strategies that learn efficiently with a goal.

In Figure 1, the ADL workflow with GPs is illustrated. The illustration is inspired and extended from [2]. The process begins with an initial DoE (step 1), where a set of initial data points is generated using sampling strategies such as Latin hypercube sampling (LHS), Sobol sequences, or space-filling designs. A GP surrogate model is then trained on this initial data set to approximate the unknown system or objective function (step 2).

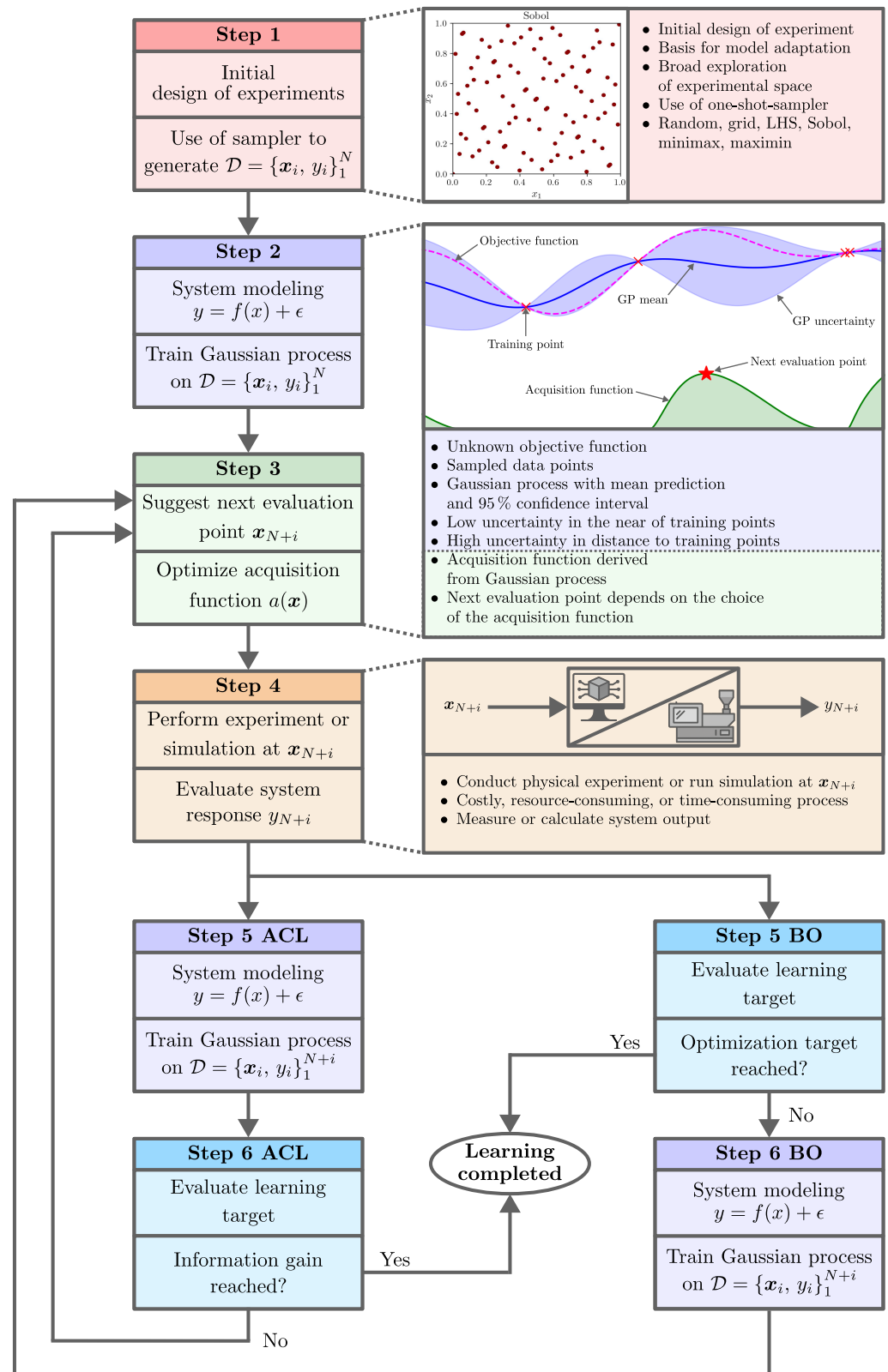


Figure 1. Adaptive learning loop with GPs.

In each iteration of the loop, the trained GP is used to propose the next evaluation point (step 3) by optimizing an acquisition function. This point is evaluated through a physical experiment or simulation (step 4), and the corresponding system response is obtained.

At this point, the workflow diverges depending on whether ACL or BO is pursued. In ACL, the GP model is retrained on the updated data set after each newly acquired data point (step 5 ACL), and the updated model is analyzed to assess whether the learning target, such as achieving a specified level of information gain or reducing predictive uncertainty, has been reached (step 6 ACL). If this is not the case, the process continues with further iterations.

Optimizing the objective function is the primary focus in BO. After each new experiment or simulation, the result is evaluated (step 5 BO) to assess whether the optimization target has been reached. If further optimization is required, the GP model is retrained (step 6 BO), and the loop continues. No explicit evaluation of a learning target is performed after each iteration in BO. Instead, the process is directly guided by the optimization objective.

Both workflows follow a similar iterative structure, yet their update and evaluation strategies differ. In ACL, the model is updated continuously, and learning progress is explicitly monitored after each new data point. In contrast, BO primarily evaluates optimization success based on the outcome of the most recent experiment or simulation, with model updates performed as needed to support ongoing optimization.

3.2. Initial Design Strategies

At the start of ADL, an initial design is required to construct the first surrogate model. These designs are often referred to as one-shot strategies in the literature [3], where all design points are selected in advance to ensure global space-filling and representative coverage of the input domain.

One of the simplest approaches is random sampling, where points x_i are drawn independently from a uniform distribution over the input space $\mathcal{X}$. While this method is flexible and easy to implement, it often leads to clustering and poor coverage in small-sample scenarios and is, therefore, mainly used as a baseline for comparison [90,91].

A more structured alternative is Latin hypercube sampling (LHS), which stratifies each input dimension into N equally sized intervals and ensures that exactly one sample is taken from each interval. This guarantees well-distributed projections along each input dimension [92], and various extensions, such as orthogonal array-based LHS (OA-LHS), further improve sampling uniformity in higher dimensions [93].

Another popular option is the use of Sobol sequences, which are low-discrepancy, quasi-random sequences that fill the space more uniformly than purely random samples. The first Sobol points in $[0, 1]^{n_x}$ have low discrepancy, leading to improved convergence in numerical integration, surrogate modeling, and sensitivity analysis [91,94].

Among geometric criteria, maximin distance designs aim to maximize the minimum distance between any pair of design points. This improves the uniform coverage of the input space, helps to avoid clustering, and enhances numerical stability in surrogate modeling [95]. Building on this idea, improved variants such as distance-distributed designs have been proposed specifically for GP surrogates [96], offering better scalability and numerical robustness in high-dimensional settings.

In contrast, minimax distance designs minimize the worst-case distance from any point in the domain to the nearest design point. This guarantees that no region of the input space is poorly represented and is particularly suited for global surrogate modeling [97].

A well-known alternative involves grid-based designs, which partition the domain into a Cartesian product of equidistant levels in each dimension. While this strategy is

suitable for low-dimensional problems [90], it suffers from exponential growth in the number of points with increasing dimension in n_x .

The full factorial design includes all possible combinations of factor levels and thus enables estimation of all main effects and interactions [98]. Although the number of runs grows rapidly, it provides a complete variance decomposition and serves as a benchmark for assessing alternative designs.

3.3. Active Learning

The process of iteratively selecting the most informative data points to improve model performance under limited data is known as active learning. In the context of GPs, this approach leverages their inherent UQ to prioritize points that reduce predictive uncertainty or improve parameter estimates. This section outlines both established and advanced acquisition strategies.

A central component of ACL is the acquisition function, which defines the learning goal and determines the next point to evaluate. The acquisition function is typically constructed from the predictive mean $\mu(x)$ and standard deviation $\sigma(x)$ of the GP. In the following, the most common acquisition functions of ACL are introduced.

3.3.1. Active Learning MacKay

One of the foundational strategies is Active Learning MacKay (ALM), introduced in [99], which aims to select points that maximize the information gain with respect to model predictions and hyperparameters. The corresponding acquisition function is defined as

$$a(x) = I(y; f, \theta \mid \mathcal{D}, x) = H(y \mid \mathcal{D}, x) - \mathbb{E}_{p(\theta \mid \mathcal{D})} \mathbb{E}_{p(f \mid \theta, \mathcal{D}, x)} (H(y \mid f, \theta, x)), \quad (9)$$

where $a(x)$ denotes the acquisition function, $H(\cdot)$ the Shannon entropy, and $\mathbb{E}_{p(\theta \mid \mathcal{D})}(\cdot)$ the expectation with respect to the posterior distribution over the GP hyperparameters $\theta \in \mathbb{R}^{N_\theta}$. The inner expectation $\mathbb{E}_{p(f \mid \theta, \mathcal{D}, x)}(\cdot)$ is taken with respect to the GP predictive distribution of the latent function $f: \mathbb{R}^{n_x} \rightarrow \mathbb{R}$ at the candidate input $x \in \mathbb{R}^{n_x}$.

The method explicitly targets information gain and is theoretically well founded, but it can exhibit undesirable behavior during early iterations. As discussed in [3], it may overemphasize noisy regions when the initial data set is uninformative, potentially leading to suboptimal acquisitions. In addition, the approach tends to favor points near the boundaries of the input space, which may contribute less to improving the model globally.

3.3.2. Fisher Information

The Fisher information (FI) criterion [3] quantifies the sensitivity of model predictions with respect to the GP hyperparameters $\theta \in \mathbb{R}^{N_\theta}$. The expected FI for a future input $x_{j+1} \in \mathbb{R}^{n_x}$ is approximated recursively as

$$\mathcal{F}_{j+1}(\theta) \approx \mathcal{F}_j(\theta) + \frac{1}{2\sigma_j^4(x_{j+1}, \theta)} \left\| \frac{\partial \sigma_j^2(x_{j+1}, \theta)}{\partial \theta} \right\|^2 + \frac{1}{\sigma_j^2(x_{j+1}, \theta)} \left\| \frac{\partial \mu_j(x_{j+1}, \theta)}{\partial \theta} \right\|^2. \quad (10)$$

Here, $\mathcal{F}_j(\theta)$ denotes the accumulated Fisher information after j observations, $\mu_j(x, \theta) \in \mathbb{R}$ is the GP predictive mean at input x given θ , and $\sigma_j^2(x, \theta) \in \mathbb{R}^+$ is the corresponding predictive variance. The derivatives $\frac{\partial \mu_j}{\partial \theta}$ and $\frac{\partial \sigma_j^2}{\partial \theta}$ quantify the sensitivity of the predictive mean and variance, respectively, with respect to the hyperparameters. This acquisition strategy emphasizes regions where the model predictions are most sensitive to changes in θ , which can improve hyperparameter estimation and thereby enhance the overall predictive

capability of the GP. However, the method may focus sampling in regions with strong gradients, potentially neglecting flat areas of the input space.

3.3.3. Bayesian Active Learning by Disagreement

Following this, Bayesian Active Learning by Disagreement (BALD) [9] is explained. The corresponding acquisition function is given by

$$a(\mathbf{x}) = H(y | \mathbf{x}, \mathcal{D}) - \mathbb{E}_{p(\boldsymbol{\theta}|\mathcal{D})}(H(y | \mathbf{x}, \boldsymbol{\theta})), \quad (11)$$

where $H(\cdot)$ is the Shannon entropy, and $\mathbb{E}_{p(\boldsymbol{\theta}|\mathcal{D})}(\cdot)$ is the expectation with respect to the posterior distribution over GP hyperparameters $\boldsymbol{\theta} \in \mathbb{R}^{N_\theta}$. The BALD acquisition function decomposes the total predictive uncertainty into an epistemic component (first term) and an aleatoric component (second term), with inputs of high epistemic uncertainty being prioritized. This makes the method particularly effective for exploring areas of the input space with high epistemic uncertainty.

3.3.4. Active Learning Cohn

Active Learning Cohn (ALC) [100,101] adopts a global perspective and selects the next point, $\mathbf{x} \in \mathbb{R}^{n_x}$, to minimize the expected global predictive variance,

$$a(\mathbf{x}) = \int_{\mathcal{X}} [\sigma_n^2(\mathbf{x}') - \tilde{\sigma}_{n+1}^2(\mathbf{x}'|\mathbf{x})] d\mathbf{x}', \quad (12)$$

where $\sigma_n^2(\mathbf{x}') \in \mathbb{R}^+$ is the GP predictive variance at location $\mathbf{x}' \in \mathbb{R}^{n_x}$ before adding the candidate point $\mathbf{x}$, and $\tilde{\sigma}_{n+1}^2(\mathbf{x}'|\mathbf{x}) \in \mathbb{R}^+$ is the predictive variance at $\mathbf{x}'$ after including $\mathbf{x}$ into the covariance structure. The integral is taken over the entire input domain $\mathcal{X}$. The expected effect of each new sample on the global uncertainty of the surrogate model is accounted for in ALC, promoting balanced sampling patterns and accurate global models. Its main drawback is the high computational cost, as the integral must be approximated for every candidate point.

By accounting for the effect of each acquisition on the entire input space, ALC tends to produce well-balanced sampling patterns. It is particularly effective for constructing accurate global surrogates and provides more stable behavior than ALM. The main drawback is its high computational cost, as the integral over $\mathcal{X}$ must be approximated for each candidate point.

3.3.5. Bayesian Query-by-Committee

A further approach is the Bayesian Query-by-Committee (B-QBC) [11], which extends the classical Query-by-Committee framework into a Bayesian setting. The method utilizes MCMC samples of the GP hyperparameters' joint posterior to draw multiple models and selects the next query point where the mean predictions of these models disagree the most. Formally, the acquisition function is defined as

$$a(\mathbf{x}) = V_{p(\boldsymbol{\theta}|\mathcal{D})}[\mu_{\boldsymbol{\theta}}(\mathbf{x}) | \boldsymbol{\theta}] = \mathbb{E}_{p(\boldsymbol{\theta}|\mathcal{D})}[(\mu_{\boldsymbol{\theta}}(\mathbf{x}) - \bar{\mu}(\mathbf{x}))^2 | \boldsymbol{\theta}], \quad (13)$$

where $\mu_{\boldsymbol{\theta}}(\mathbf{x})$ denotes the GP predictive mean given hyperparameters $\boldsymbol{\theta}$, and $\bar{\mu}(\mathbf{x})$ is the posterior average mean. The B-QBC strategy prioritizes points where sampled models exhibit strong disagreement, which typically corresponds to regions of high posterior variance. Since the models are drawn from the hyperparameters' posterior, the method implicitly emphasizes posterior modes and can be interpreted as a mode-seeking acquisition strategy.

3.3.6. Query by Mixture of Gaussian Processes

Another extension is the Query by Mixture of Gaussian Processes (QB-MGP) [11], which accounts not only for model disagreement but also for predictive uncertainty. Using N_{Post} MCMC samples of the hyperparameters, the predictive posterior is represented as a mixture of Gaussian processes with mean

$$\mu_{\text{GMM}}(\mathbf{x}) = \frac{1}{N_{\text{Post}}} \sum_{j=1}^{N_{\text{Post}}} \mu_{\theta_j}(\mathbf{x}), \quad (14)$$

and variance

$$\sigma_{\text{GMM}}^2(\mathbf{x}) = \frac{1}{N_{\text{Post}}} \sum_{j=1}^{N_{\text{Post}}} \sigma_{\theta_j}^2(\mathbf{x}) + \frac{1}{N_{\text{Post}}} \sum_{j=1}^{N_{\text{Post}}} \left(\mu_{\theta_j}(\mathbf{x}) - \mu_{\text{GMM}}(\mathbf{x}) \right)^2. \quad (15)$$

The resulting acquisition criterion is defined as

$$a(\mathbf{x}) = \mathbb{E}_{p(\theta|\mathcal{D})} \left[\sigma_{\theta}^2(\mathbf{x}) \mid \theta \right] + \mathbb{E}_{p(\theta|\mathcal{D})} \left[\left(\mu_{\theta}(\mathbf{x}) - \mu_{\text{GMM}}(\mathbf{x}) \right)^2 \mid \theta \right], \quad (16)$$

which simultaneously combines predictive variance (as in B-ALM) and model disagreement (as in B-QBC). The QB-MGP strategy, therefore, favors regions with both high predictive uncertainty and high model disagreement, yielding a balanced exploration pattern.

3.3.7. Residual Active Learning

A residual-based criterion ranks candidates by their current prediction error with respect to available reference responses. A common score is the absolute or squared residual,

$$a(\mathbf{x}) = |y(\mathbf{x}) - \mu(\mathbf{x})| \quad \text{or} \quad a(\mathbf{x}) = (y(\mathbf{x}) - \mu(\mathbf{x}))^2, \quad (17)$$

optionally standardized by the predictive standard deviation. The samples with the largest residuals are iteratively added until the validation metric stabilizes, as used in [102].

3.3.8. Euclidean Distance-Based Diversity

A diversity-based criterion adds candidates that are far from the current training set in feature space. The maximin score is

$$a(\mathbf{x}) = \min_{\mathbf{x}_i \in \mathcal{X}} \|\mathbf{x} - \mathbf{x}_i\|_2, \quad (18)$$

and a greedy farthest-point strategy increases coverage while reducing redundancy, as used in [102].

3.3.9. Integrated Mean Squared Prediction Error

Following [3], the Integrated Mean Squared Prediction Error (IMSPE), also written as IMS(P)E when predictive is omitted, averages the mean squared prediction error of the observed output over the domain. For GP regression, the pointwise MSPE equals the predictive variance,

$$\text{MSPE}_n(\mathbf{x}) = \mathbb{E} \left[(\mu(\mathbf{x}) - y(\mathbf{x}))^2 \mid \mathcal{D} \right] = \sigma_{n,Y}^2(\mathbf{x}), \quad (19)$$

so the acquisition integrates the observed predictive variance (optionally with a weight w)

$$a(\mathbf{x}) = \int_{\mathcal{X}} \tilde{\sigma}_{n+1}^2(\mathbf{x}' \mid \mathbf{x}) w(\mathbf{x}') d\mathbf{x}'. \quad (20)$$

The IMSPE promotes global coverage and reduces overall predictive uncertainty. The drawback is the computational cost of evaluating the integral for each candidate.

3.3.10. Sensitivity-Based Active Learning

A recent complementary strategy is Sensitivity-Based Active Learning (SBAL) [103], which leverages variance-based sensitivity analysis (SA) to guide acquisitions. The acquisition function is implicitly defined by selecting the input $\mathbf{x} \in \mathbb{R}^{n_x}$ that maximizes the expected predictive standard deviation in the most sensitive input dimensions:

$$a(\mathbf{x}) = \mathbb{E}_S[\sigma(\mathbf{x})], \quad (21)$$

where $\sigma(\mathbf{x}) \in \mathbb{R}^+$ is the GP predictive standard deviation at $\mathbf{x}$, and $\mathbb{E}_S[\cdot]$ represents the expectation weighted by the sensitivity coefficients $S \in \mathbb{R}^{n_x}$. These sensitivity weights are derived from Sobol indices [104], which quantify the relative contribution of each input dimension to the variance of the predictive distribution. After training the GP on an initial data set, first-order Sobol indices $S_{\sigma 1}$ and second-order Sobol indices $S_{\sigma 2}$ are computed for the predictive standard deviation, identifying the most influential input dimensions. Candidate points in areas of the input space that exhibit both high predictive uncertainty and high sensitivity to influential inputs are prioritized by SBAL. This makes SBAL particularly effective for problems with heterogeneous input relevance or partially structured input spaces.

3.4. Bayesian Optimization

Bayesian optimization [2] is particularly suited to applications where each evaluation of the objective function is costly or time-consuming, such as in the optimization of chemical products and functional materials [6,105], hyperparameter optimization of ML models [7,106,107], or complex engineering simulations. In this setting, the objective function is typically unknown and derivative-free, accessible only through point-wise evaluations.

To model the objective function and quantify epistemic uncertainty, a probabilistic surrogate model is employed, most commonly a GP [10] or a Tree-structured Parzen Estimator (TPE) [108]. The surrogate provides both a predictive mean and an uncertainty estimate, which in turn enable the construction of an acquisition function that guides the selection of new query points. The adaptive sampling strategy of the acquisition function dynamically balances the exploration of under-sampled areas of the input space with the exploitation of currently promising areas [70]. While standard BO is particularly effective in low-dimensional spaces (typically up to about 15 input variables), specialized kernel methods and dimensionality reduction techniques have been developed to extend its applicability to higher-dimensional problems [2].

3.4.1. Acquisition Functions

A central component of BO is the acquisition function, which defines the trade-off between exploration and exploitation and determines the next point to evaluate. The acquisition function is typically constructed from the predictive mean $\mu(\mathbf{x})$ and the standard deviation $\sigma(\mathbf{x})$ of the GP. Its global maximizer is chosen as the next evaluation point, commonly using multi-start local optimizers, such as the L-BFGS optimizer [109], or stochastic optimizers like Nadam [110].

Expected Improvement

One of the most widely used acquisition functions is the Expected Improvement (EI) [4], which quantifies the expected improvement over the current best observation. The corresponding acquisition function is defined as

$$a(x) = (\mu(x) - f(x^*) - \rho) \Phi\left(\frac{\mu(x) - f(x^*) - \rho}{\sigma(x)}\right) + \sigma(x) \varphi\left(\frac{\mu(x) - f(x^*) - \rho}{\sigma(x)}\right), \quad (22)$$

where $f(x^*)$ is the current best observation, $\Phi(\cdot)$ is the cumulative distribution function of the standard normal distribution, $\varphi(\cdot)$ is its probability density function, and ρ is an optional exploration parameter. The method provides a natural trade-off between exploration and exploitation. However, it can become overly exploitative if ρ is not tuned appropriately.

Probability of Improvement

A simpler alternative is the Probability of Improvement (PI) [4], which maximizes the probability of achieving an improvement over the current best observation. The corresponding acquisition function is defined as

$$a(x) = \Phi\left(\frac{\mu(x) - f(x^*) - \rho}{\sigma(x)}\right). \quad (23)$$

This approach is computationally efficient but tends to focus excessively on exploitation unless a significant exploration parameter, ρ , is introduced.

Upper/Lower Confidence Bound

The Upper Confidence Bound (UCB) [4] selects points that maximize a weighted combination of the predictive mean and uncertainty. The acquisition function is given by

$$a(x) = \mu(x) + \kappa \sigma(x), \quad (24)$$

where κ is a user-defined parameter that controls the exploration–exploitation balance. The Lower Confidence Bound (LCB) uses the lower band of the predictive distribution with

$$a(x) = \mu(x) - \kappa \sigma(x), \quad (25)$$

and is likewise maximized for a given κ .

Thompson Sampling

Another widely used strategy is Thompson Sampling (TS) [2], which selects the next point by sampling a function realization $f^{(s)}(x)$ from the posterior distribution of the surrogate model and choosing the maximizer as

$$a(x) = f^{(s)}(x). \quad (26)$$

This method naturally balances exploration and exploitation and is simple to implement, though it may require many posterior samples to converge in highly multi-modal landscapes.

Knowledge Gradient

The Knowledge Gradient (KG) [2] explicitly models the expected value of information gained by sampling a point. It is defined as

$$a(x) = \mathbb{E}\left[\max_{x'} \mu^{(x)}(x')\right] - \max_{x'} \mu(x'), \quad (27)$$

where $\mu^{(x)}$ is the updated predictive mean after observing the result at x . This acquisition function is particularly effective in noisy or expensive settings, as it anticipates the impact of new data on the model's knowledge of the optimum.

(Predictive) Entropy Search

Entropy Search (ES) and its variant Predictive Entropy Search (PES) [2] aim to reduce the uncertainty about the location of the global optimum. The acquisition function is based on the mutual information between the new observation and the unknown global optimum x^* ,

$$a(x) = H(p(x^* | \mathcal{D})) - \mathbb{E}_{y(x)}[H(p(x^* | \mathcal{D} \cup \{(x, y(x))\}))]. \quad (28)$$

These methods are powerful for fully global optimization but are computationally expensive due to the need for entropy estimation and posterior sampling.

3.4.2. Batch Bayesian Optimization

In scenarios where parallel evaluations are possible, batch BO is employed to select multiple points per iteration [2]. This is particularly relevant in experimental settings where conducting only a single experiment per iteration would be inefficient, for example, when using robotic experimental platforms or running parallel experiments on physical systems such as chemical reactors, materials synthesis stations, or high-throughput screening equipment.

In batch settings, acquisition functions must be adapted to account for interactions between points within a batch. Extensions of EI and UCB to batch settings include the q Expected Improvement (q EI) and the q Upper Confidence Bound (q UCB) [111], which jointly optimize the acquisition function over a batch of q points.

Alternatively, local penalization methods adjust the acquisition function to penalize regions around already selected batch points, promoting diversity in the batch. Information-theoretic approaches such as BatchBALD [9] select batch points by maximizing the joint mutual information between the batch and the model parameters or the predictive target, further encouraging both diversity and informativeness. The batch BO approach is particularly advantageous in parallel computing environments or when experimental platforms are capable of evaluating multiple configurations simultaneously, thereby significantly accelerating the optimization process.

3.4.3. Multi-Goal Bayesian Optimization

Many real-world optimization problems involve multiple, potentially conflicting objectives [2]. In such cases, multi-objective Bayesian optimization (MOBO) aims to approximate the Pareto front of optimal trade-offs. One common approach is to scalarize multiple objectives into a single objective using a weighted sum. However, scalarization may fail when objectives are strongly conflicting or non-convex. Pareto-based methods instead aim to directly identify Pareto-optimal points, where no objective can be improved without degrading another.

Predictive Entropy Search can be extended to MOBO, allowing scalable optimization by decoupling objectives and ensuring that computational cost grows linearly with the number of objectives [2].

A key challenge in MOBO is the increasing computational cost as the number of objectives grows. Recent research focuses on designing more scalable acquisition functions to address this limitation. Furthermore, it is important to distinguish MOBO from multi-task BO. In multi-task settings, a family of related optimization tasks is modeled jointly, leveraging shared information to improve sample efficiency and predictive accuracy. Both

MOBO and multi-task BO are applied in domains such as materials discovery, robust engineering design, and trade off aware learning systems [2].

3.5. Overview of Adaptive Learning Methods

Adaptive learning combines diverse sampling and acquisition strategies to efficiently guide data collection. These strategies are commonly grouped according to the phase of the learning process. Initial design strategies are used to generate an informative starting data set for surrogate model training, active learning strategies are employed to iteratively select points to improve model accuracy, and Bayesian optimization strategies are applied to locate the global optimum of an unknown objective function.

In Table 2, the reviewed methods for ADL are summarized across these categories, highlighting their key characteristics, limitations, and representative references. The table provides a compact overview of the methodological landscape discussed in this section and serves as a reference for selecting appropriate strategies, depending on the problem context and learning goals. Within each category, the methods are ordered by the publication year.

Table 2. Overview of sampling and acquisition strategies in ADL.

Category	Method	Characteristics	Limitations	References
Initial Design	Random Sampling	Simple, flexible, baseline	Clustering, poor space-filling	[90,91]
Initial Design	Latin hypercube sampling (LHS)	Stratified, well-distributed projections	Axis-aligned bias, lacks optimality guarantees	[92,93]
Initial Design	Sobol sequence	Low-discrepancy, quasi-random, excellent space-filling	Axis bias for small n , deterministic	[91,94]
Initial Design	Maximin distance	Maximizes minimum distance, uniform dispersion	Computationally expensive optimization	[95,96]
Initial Design	Minimax distance	Guarantees uniform global coverage	Expensive in high dimensions	[97]
Initial Design	Grid/full factorial design	Exhaustive coverage of factor combinations	Exponential growth with dimension	[90,98]
Active Learning	Active Learning MacKay (ALM)	Maximizes information gain, theoretical grounding	Sensitive to noise, boundary bias	[3,99]
Active Learning	Fisher Information (FI)	Targets hyperparameter-sensitive regions	Focuses on gradient regions, neglects flat areas	[72]
Active Learning	Bayesian Active Learning by Disagreement (BALD)	Explores poorly understood regions	Requires posterior sampling, sensitive to surrogate quality	[9]
Active Learning	Active Learning Cohn (ALC)	Minimizes global predictive variance	High computational cost (global integral)	[100,101]
Active Learning	Bayesian Query-by-Committee (B-QBC)	Posterior-based model disagreement	GP predictive uncertainty not considered	[11]
Active Learning	Query by Mixture of Gaussian Processes (QB-MGP)	Mixture of GP models, combines disagreement and variance	No explicit balance of epistemic vs. aleatoric uncertainty	[11]
Active Learning	Residual Active Learning (RSAL)	Ranks candidates by residual error, focuses mispredictions	Requires reference labels, sensitive to noise	e.g., [102]
Active Learning	Euclidean Distance-Based Diversity (EBD)	Adds farthest points in feature space, improves coverage	Ignores model uncertainty, boundary bias	e.g., [102]
Active Learning	IMSE	Minimizes integrated posterior variance, global coverage	High computational cost (domain integral)	[3]
Active Learning	SBAL	Targets sensitive and uncertain dimensions, fast convergence	Requires sensitivity analysis	[104]

Table 2. Cont.

Category	Method	Characteristics	Limitations	References
Bayesian Optimization	Expected Improvement (EI)	Balances exploration/exploitation	Tends to over-exploit if ρ not tuned	[4]
Bayesian Optimization	Probability of Improvement (PI)	Simple, efficient	Exploitative, sensitive to ρ	[4]
Bayesian Optimization	Upper/Lower Confidence Bound (UCB, LCB)	Theoretically founded, parameter-controlled trade-off	Choice of κ critical	[4]
Bayesian Optimization	Thompson Sampling (TS)	Posterior sampling, balances exploration/exploitation	Needs many samples for multi-modal functions	[2]
Bayesian Optimization	Knowledge Gradient (KG)	Explicit value of information, anticipates model improvement	Computationally intensive	[2]
Bayesian Optimization	Entropy Search (ES)/PES	Reduces uncertainty on global optimum location	High computational cost, entropy estimation required	[2]
Bayesian Optimization	Batch BO (qEI, qUCB, BatchBALD)	Enables parallel experiments, batch diversity	Interaction effects in batch optimization	[2,9,111]
Bayesian Optimization	Multi-Objective BO (MOBO)	Pareto optimization, multiple objectives	Scalability with objectives, complex trade-offs	[2]

4. Applications of Adaptive Learning with Gaussian Processes

In this section, applications of ADL with GP regression over the last five years are analyzed, and a small number of earlier works are included where the contributions are fundamental and widely cited in the literature. In Table 3, the search strategy together with the corresponding database-specific query results and inclusion counts is summarized. In Table 4, a compact map from application context to methodological choices is shown. Within each application category, the methods are ordered by the publication year. The columns list application field, learning goal, data type, learning type, initialization strategy, acquisition function, GP model class, and reference. Each row summarizes one published study. Multiple entries within a cell indicate combined settings such as “Fcn., Sim.” or “EI, UCB”. Rows that begin with “Methods/ General” summarize purely methodological work that is evaluated on benchmark functions or simulation studies without a specific application field, whereas all other entries starting with “Methods/...” indicate methodological contributions that are validated in an application. The table serves as the basis for the following analysis, where defaults are identified, exceptions are highlighted, and gaps that limit the deployment of ADL in real-world applications are discussed.

4.1. Methodology of the Literature Search for Application Studies

In this subsection, the methodology of the literature search for the application-focused corpus is described to improve transparency of identification, screening, and final selection. The reporting structure is organized in a PRISMA-oriented manner, following the Preferred Reporting Items for Systematic Reviews and Meta-Analyses (PRISMA) 2020 statement to document identification, screening, eligibility assessment, and inclusion decisions for the application-oriented studies in a transparent and reproducible way [112]. The search was focused specifically on practical applications of GP-based ADL and on the studies summarized in Table 4. The broader review additionally includes seminal contributions, highly cited methodological works, and a limited number of very recent papers that are important for positioning the field.

Two bibliographic databases were used, namely Scopus and Web of Science. Both are widely used multidisciplinary bibliographic databases, and differences in indexing scope, included sources, and coverage characteristics have been reported in the literature [113,114].

The combined use of Scopus and Web of Science was intended to reduce the risk of missing relevant studies due to database-specific coverage differences and to mitigate single-database bias during identification. A consolidated overview of database-specific query results and included primary studies is provided in Table 3. Fixed query blocks and fixed filters were used, and queries were executed via database APIs. The search was based on a GP block and on an ADL block.

For Scopus, the initial retrieval returned 3333 records. Manual title and abstract screening of the resulting application-focused corpus yielded 30 studies represented in Table 4.

For Web of Science, the initial retrieval returned 2422 records. Manual title and abstract screening identified 24 studies represented in Table 4. Deduplication against the Scopus-included set showed that these studies overlap with the Scopus application corpus, such that no additional unique application study was contributed by Web of Science for Table 4.

The resulting database-screened corpus, therefore, constitutes a transparent application-focused basis for the practical studies discussed in this section. At the same time, the broader review also includes additional targeted references that reflect domain knowledge accumulated over multiple years of research in GPs and ADL, including seminal works, highly cited methodological contributions, and very recent papers that are important for positioning the field. Coverage limitations and indexing differences can still lead to omissions despite the dual-database strategy [113,114].

Table 3. Search queries and inclusion results for Scopus and Web of Science.

Database	Link	Query	Total	Relevant	Included	Excluded
Scopus	https://www.scopus.com/search/form.uri?display=advanced (accessed on 23 March 2026)	TITLE-ABS-KEY((("gaussian process*" or kriging) and ("active learning" or "adaptive learning" or "adaptive sampling" or "sequential design" or "bayesian optimi*" or "efficient global optimi*" or "safe exploration" or "information gathering"))) and PUBYEAR > 2003 and PUBYEAR < 2026)	3333	30	30	3303
Web of Science	https://www.webofscience.com/wos/woscc/advanced-search (accessed on 23 March 2026)	TS=((("gaussian process*" or kriging) and ("active learning" or "adaptive learning" or "adaptive sampling" or "sequential design" or "bayesian optimi*" or "efficient global optimi*" or "safe exploration" or "information gathering")))) and PY=(2004-2025)	2422	24	0	2422
Fused result					30	

4.2. Fields of Application

In this subsection, the use of ADL with GP regression across major application fields is summarized; as well, typical patterns and gaps in each domain are highlighted.

4.2.1. Aerospace

In aerospace, most studies are simulation-driven and aim at accurate global surrogates that can replace expensive solvers. In this application field, ACL with global variance reduction is commonly used [13,54]. As shown in Table 4, FI-based criteria are used for surrogate modeling of composite fuselage shape control [15]. Non-stationary kernels, local

experts, and partitioned designs are reported more often than in other domains, which aligns with heterogeneous flow physics and localized features [13,54,115,116].

4.2.2. Chemical Engineering

In chemical engineering applications, BO is reported as the default for experimental studies with space-filling starts. This pattern is reflected by coating and reaction studies that adopt LHS or Sobol for initialization and EI or UCB during optimization [88,105,117]. Multi-objective settings are frequent, and multi-output GPs or explicit multi-objective criteria are used when several quality metrics have to be optimized together [84,118]. Advanced non-stationary models are rarely used in experimental use-cases, while real experimental loops continue to favor a standard GP [88]. The limited use of heteroscedastic likelihoods and outlier-robust noise models can be identified despite input-dependent variability and occasional outliers in practice.

4.2.3. Dynamic Systems

In dynamic systems, ADL is used to tune controllers and to emulate dynamics. Batch BO is applied to MPC parameter tuning, and heteroscedastic GPs are used to capture regime-dependent variability [119]. Surrogate construction for dynamical models with ACL is reported, and practical controller tuning with BO on experimental setups is present [120,121]. Closed-loop safe exploration, online updating with stability guarantees, and consistent physics constraints in both model and acquisition remain open points for wider adoption.

4.2.4. Geoscience and Environmental Monitoring

In geoscience and environmental monitoring, GP regression is used to retrieve biophysical variables from aerial and satellite imagery. Training samples are selected with diversity and uncertainty criteria within ACL workflows to reduce required reference measurements and improve coverage [102,122]. Both studies analyze per-pixel predictive uncertainty and rely on standardized benchmark data sets [102,122].

4.2.5. Manufacturing

In manufacturing, the standard GP is used in simulation studies and in real experiments across welding, laser processing, printed parts, and process parameter tuning. Here, EI is the common choice, and UCB or LCB appear when a tunable margin is preferred [123–127]. In simulation-based use-cases, deep GPs are explored for process parameter optimization with BO but are not the common choice in operational loops [128]. Constraint handling through constrained EI and feasibility rules is reported but is not yet standardized as part of an automated ADL loop. Safe limits are frequently enforced by domain experts outside the algorithm [126,129]. Across the manufacturing entries in Table 4, three recurring gaps can be identified. In multi-task settings for manufacturing, IMSE-based criteria are used to guide sampling with multi-task GP surrogates [71]. The systematic modeling of input-dependent noise is missing, non-stationary cross-covariances for coupled quality metrics are completely absent, and modeling that injects safe process windows directly into the acquisition is not routine [129].

4.2.6. Materials Science

In materials science, BO is used for design targets [130], and ACL is used for surrogate modeling of material properties [131]. Partitioned models and acquisitions are employed in simulations when heterogeneous responses are expected [131]. Multi-output GPs are applied when several material properties have to meet joint targets [125]. Open issues include non-stationary cross-covariances for multi-output ADL and broader use in real experimental studies.

4.2.7. Robotics

In robotics, feasibility and robustness are emphasized. Constrained and safe optimization is adopted when safety boundaries are tight, and robust EI is used to handle disturbances [83,132]. Multi-output models are used when control and task metrics interact [133]. Information gathering and mapping tasks employ ACL with domain-driven pilot paths and non-stationary kernels where occlusions or discontinuities occur [53]. Practical limits arise from time budgets for acquisition optimization and observation noise that is not well described by a GP model.

4.2.8. Structural Reliability

In structural reliability, probability of failure and behavior near limit states are prioritized. After a space-filling start, ACL is used to refine the surrogate near the failure region [134]. Heteroscedastic noise models are common and often necessary [135,136]. Multi-output GPs are applied when several responses or load channels have to be predicted together [134]. Acquisitions that move the design toward the failure boundary while limiting extrapolation risk are reported, and batch ACL appears underused [137,138].

4.3. Methodologies

In this subsection, methodological work and cross-cutting patterns in model design, multi-output usage, initial designs, and acquisition functions are summarized.

4.3.1. Method-Oriented Studies

In method-oriented studies, non-stationary and local models are evaluated mainly on functions and simulations. Representative models include warped multiple-index GPs, HHK-GP, PCEGP, DGP, JGP, and DJGP for non-stationary behavior [12,13,45,54,139]. The acceleration of BO and the use of known problem structure are central themes. Scaling can be achieved with Vecchia-style approximations for batch BO [78]. On function networks, the known computational graph is exploited to decide which node to sample and at which inputs. Recent methods enable cost-aware partial evaluations by querying only a subset of nodes when this is expected to improve the overall objective efficiently [140,141]. Committee- and mixture-based acquisition strategies such as B-QBC and QB-MGP are designed to balance bias and variance [11]. Consistent gains are shown in controlled simulations, while transfer to industrial ADL loops is still limited, as indicated by the entries in Table 4. Many ADL studies already address expensive simulations, where surrogate models are used to reduce the cost of repeated evaluations. A further practical challenge arises when simulation-based learning must be transferred to physical experiments, which are often even more costly and may deviate systematically from the simulated response. In such settings, multi-fidelity GP frameworks are relevant because they allow lower-cost simulations to provide broad structural information, while a limited number of experiments is used to correct discrepancies and calibrate the surrogate to physical reality. More generally, this can also be interpreted from a transfer-learning perspective, where information from simulations is reused to improve learning efficiency in the experimental target domain.

4.3.2. Multi-Output Usage

Across applications with multiple coupled objectives or sensors, multi-output modeling is widely used. Shared structure reduces variance and supports consistent comparison between objectives [84,125,142]. In Table 4, it is also shown that ACL is still configured as a single-output procedure in many studies. This indicates a gap because many surrogate-building tasks could benefit from multi-output acquisitions that reflect coupled objectives, constraints, and shared physics.

4.3.3. Initial Designs

Across domains, space-filling initial designs are used by default. A space-filling start with LHS is often selected because good projection properties are obtained with low setup effort [54,102,105,117,131,134,136,139]. When more uniform projections across coordinates are required or when higher dimensions are present, Sobol sequences are preferred [78,88,123,125,135]. Starts that follow domain knowledge are employed when safe regions or predefined trajectories exist [53,83,127,143,144]. The pattern observed in Table 4 confirms this practice. These choices set early hyperparameter scales and influence exploration and stability.

4.3.4. Acquisition Function Usage over Domains

Across applications, EI is the most frequently used BO acquisition in both simulation and experimental studies [105,117,123,125,126,145]. Confidence bound criteria, such as the UCB and LCB, are used when a tunable exploration margin is preferred or when conservative moves are desired [88,127]. Multi-objective settings use Expected Hypervolume Improvement (EHVI) and related criteria [118]. The use of KG appears in [141] on function-network. In surrogate construction under ACL, variance-based criteria are used most often, with ALM reported as the common choice [12,13,139,143]. Global variance reduction with ALC is described next in frequency and is applied when broad coverage of the design space is targeted [45,131,134,139]. The use of IMSPE is documented in multi-output settings as an exception within ACL, since single-point acquisitions remain the norm [71]. In Bayesian inverse problems, goal-oriented sequential design is formulated within Stepwise Uncertainty Reduction (SUR), where candidate points are chosen to reduce an inversion-relevant uncertainty measure. Representative criteria include Constraint Set Query (CSQ) and Inverse Problem Stepwise Uncertainty Reduction (IP-SUR) [142]. These observations indicate a gap for ACL acquisition functions in batch and multi-output formulations that could be addressed in future work. Diversity-based acquisitions and residual-based rules are used for sensor placement and plant monitoring [102]. In safety-critical and disturbance-prone studies, safe exploration and feasibility awareness are addressed with methods such as SafeOpt, constrained EI, and robust EI [83,126,132]. In SafeOpt, a GP-based feasibility model maintains a conservative safe set, and candidates are proposed within this set [83]. Batch querying is widely used in BO loops and appears less often in ACL, even though parallel computing is available [84,88,105,146].

4.4. Gaps and Guidance for Practical Application

Advanced non-stationary and local GP methods are rarely integrated into industrial ADL loops. The corresponding benefits are demonstrated mainly in simulations and controlled benchmarks [12,13,139]. Without informative priors, identification of output scale, lengthscale, and noise is difficult under small data sets, and repeated retuning increases operational effort. Systematic use of input-dependent noise models is limited outside structural reliability, even though many industrial data sets show input-dependent variability [135,136]. Constraint-aware and safe acquisitions are reported but are not a default choice in routine experimentation [83,126]. Batch ACL for global surrogates appears underrepresented, and credible approximations to ALC and FI are needed for large candidate sets. Non-stationary cross-covariances in multi-output models are rarely used in practical studies. The use of global non-stationary GP models that embed input-dependent hyperparameters in a single global kernel, rather than relying on local or partitioned experts, is also uncommon in practice and has not yet seen broad adoption in industrial ADL loops. The industrial deployment of non-stationary GP models more generally is

still limited. Posterior multi-modality is observed in [11,54] and can be addressed with tempered transitions.

A practical path for industrial ADL can be stated as follows. A standard GP should be retained as the default model in the loop because training is reliable, tuning is comparatively simple, and the maintenance risk is low. More complex models should not be introduced by default but only when diagnostics indicate that the assumptions of the stationary baseline are no longer adequate. In practice, this can be assessed through residual patterns, systematic local misfit, calibration deficiencies, or clearly input-dependent variance structures. Advanced models, such as non-stationary or heteroscedastic GPs, can then be trained in parallel and evaluated on held-out or cross-validation data. The default model should only be replaced as the acquisition driver when lower error and improved calibration are observed consistently. Otherwise, the standard GP can remain in place. Region-specific non-stationarity can be introduced only where residual diagnostics repeatedly indicate localized violations of stationarity, such as structured local misfit or systematic changes in smoothness across the input domain. Input-dependent noise models can be added when variance patterns with inputs are evident. Feasibility surrogates and conservative sets can be used to integrate process windows into the acquisition, with constrained EI and safe exploration. Batch ACL could be adopted more widely and paired with scalable approximations to ALC or FI to exploit parallel resources. Multi-output acquisitions could be used when objectives are coupled. Multi-output cross-covariances with non-stationary structure could be introduced when outputs are coupled, and the relationships vary across the input space, using shared and task-specific kernels or partitioned structures with identifiable parameterizations and credible priors.

For industrial deployment, explainability should not be limited to the GP model alone, but should extend to the ADL loop as a whole. Trust in such loops is strengthened when model choice is transparent, residual diagnostics are monitored explicitly, acquisition decisions are traceable, and uncertainty contributions can be interpreted in a structured way. In this sense, explainable ADL loops should make it possible to understand why a particular model is retained or replaced, why a candidate point is selected, and whether uncertainty is driven mainly by sparse data, model misspecification, or noise. Such transparency is particularly important when more complex non-stationary, multi-output, or approximation-based GP models are introduced in practical workflows.

4.5. Overview of GP-Based ADL Applications

In Table 4, the reviewed applications of GP-based ADL are summarized across domains, workflow types, initialization strategies, acquisition functions, and GP model classes. Each row represents one reviewed study, thereby providing a structured overview of the practical application landscape discussed in this section. Within each application category, the studies are ordered by publication year.

In Figure 2, broad temporal patterns within the reviewed application corpus from 2021 to 2025 are summarized. It is indicated that the number of reviewed GP-based ADL applications increases overall across this period, while both standard GP applications and applications using advanced GP methods remain represented. It is also shown that applications using ACL-related and BO-related settings continue to appear throughout the reviewed corpus. The visualization is intended to illustrate broad trends within the reviewed studies rather than to provide a fully exhaustive bibliometric analysis of the field.

Table 4. Overview of practical applications of GP-based ADL.

Application	Goal	Data	Type	Init.	Acq.	GP Model	References
Aerospace	Wing shape design of a UAV	Sim.	BO	LHS	EI	Comb. global/local GPs	[115]
Aerospace	Surrogate modeling of shape control of composite fuselage	Sim.	ACL	Maximin LHS	Var., FI	Multiple GPs	[15]
Aerospace	Optimization of non-stationary aerospace problems	Fcn., Sim.	BO	LHS	EI	Deep GP	[116]
Gaps	simulation-driven, validation in real experimental loops not reported						
Chemical	Optimization of coating properties	Exp.	Batch BO	LHS, full factorial	EI	DGCN, non-stationary	[105]
Chemical	Optimization of chemical reactions and material consumption	Exp.	BO	LHS	EI	Multi-output GP	[117]
Chemical	Optimization of coating properties	Exp.	Batch BO	Sobol	UCB	GP	[88]
Chemical	Optimization of catalyst properties for higher alcohol synthesis	Exp.	Batch BO	-	EI, EHVI, ALM	GP	[118]
Chemical	Surrogate modeling of pharmaceuticals	Exp.	Batch ACL	-	ALM	GP	[147]
Chemical	Acceleration of automated discovery of drug molecules	Sim., data sets	Batch BO	Random	EI	Multi-output GP	[84]
Gaps	Advanced non-stationary models are rarely used in experimental use-cases, heteroscedastic likelihoods and outlier-robust noise models are not used						
Dynamic systems	Surrogate modeling of dynamic systems	Sim.	ACL	-	ALM	GP	[120]
Dynamic systems	Optimization of MPC-parameter	Sim.	Batch BO	Zero initializing	EI	GP with heteroscedastic noise	[119]
Dynamic systems	Optimization of control parameters	Exp.	BO	Grid	EI	GP	[121]
Gaps	Closed-loop safe exploration, online updating with stability guarantees, consistent physics constraints in both model and acquisition						
Environment monitoring	Optimization of measurement locations to minimize model uncertainty	Sim.	BO	Random	EI	GP	[85]
Environment monitoring	Surrogate modeling of plant growth	Sim., Exp.	ACL	LHS	RSAL, EBD	GP	[102]
Geoscience	Surrogate modeling of essential climate variables	data set	ACL	Random	Div., Var.	GP	[122]
Gaps	Studies rely on simulations and data sets, operational deployment aspects are not addressed						
Manufacturing	Constrained optimization of process parameters for turning	Sim., Exp.	BO	Random, Grid	Constrained EI	GP	[126]
Manufacturing	Surrogate modeling of surface shapes	data set	ACL	Random	IMSE	Multi-task GP	[71]
Manufacturing	Learn countersink depths	Exp.	ACL	Process-driven	ALM	GP	[143]
Manufacturing	Optimization of welding parameters	Exp.	BO	Sobol	EI	GP	[123]
Manufacturing	Optimization of laser power profile	Sim.	BO	LHS	UCB	GP	[124]
Manufacturing	Optimization of manufacturing parameters	Fcn., Sim.	BO	LHS	EI	Deep GP	[128]
Manufacturing	Optimization of parameters for laser power control	Exp.	BO	Predefined	LCB	GP	[127]
Manufacturing	Constraint learning for manufacturing process design	Fcn., Exp.	Batch ACL, BO	LHS	EI, PI, TS, UCB	GP	[129]
Manufacturing	Optimization of shape accuracy in 3D print	Exp.	Batch BO	Sobol	EI	Multi-output GP	[125]
Gaps	Systematic modeling of input-dependent noise is missing, non-stationary cross-covariances for coupled quality metrics are completely absent, and modeling that injects safe process windows directly into the acquisition is not routine						

Table 4. Cont.

Application	Goal	Data	Type	Init.	Acq.	GP Model	References
Materials science	Optimization of parameters for material design	Sim.	BO	-	-	GP	[130]
Materials science	Surrogate modeling of corrosion-resistant alloy design	Fcn., Sim.	ACL	LHS	Partitioned ALC	Partitioned GP	[131]
Material science	Optimization of the insulation coating process for alloy sheets	Exp.	BO	-	EI	GP	[145]
Materials science	Surrogate modeling for identifying fissile material	Fcn., Sim.	ACL	LHS	CSQ, IP-SUR	Multi-output GP	[142]
Gaps	Non-stationary cross-covariances for multi-output ADL, broader use in real experimental studies remains limited						
Robotic	Verification of complex safety specifications	Sim.	BO	Random	LCB	GP	[148]
Gaps	simulation-driven, validation in real experimental loops not reported						
Structural reliability analysis	Surrogate modeling of structural reliability analysis	Sim.	ACL	LHS	U-, EFF-, H-fcn.	Multi-output GP	[134]
Structural reliability analysis	Surrogate modeling of structural reliability of wind-excited systems	Sim.	BO	LHS	Failure criterion	GP with heteroscedastic noise	[136]
Structural reliability analysis	Surrogate modeling of structural reliability analysis of airplane parts design	Fcn., Sim.	ACL	Sobol	Weighted Error and Uncertainty	GP with heteroscedastic noise	[135]
Gaps	Batch ADL underused, simulation-driven, validation in real experimental loops not reported						
Methods/General	Surrogate modeling	Sim.	ACL	LHS	ALM	Bayesian Treed GP	[42]
Methods/General	Surrogate modeling	Sim.	ACL	LHS	MSPE	Local GPs	[72]
Methods/General	Surrogate modeling with avoiding critical regions	Fcn., Sim.	ACL	Safe sampling points	ALM	GP	[144]
Methods/General	Surrogate modeling	Fcn., Sim.	ACL	Maximin LHS	ALM	Warped Multiple Index GP, non-stationary Multi-output GP	[12]
Methods/General	Optimization of computational complexity in multi-objective BO	Sim.	Batch BO	Random	EI, EHVI	GP	[149]
Methods/General	Surrogate modeling of structural reliability analysis	Sim.	ACL	LHS	Conditional likelihood	GP	[150]
Methods/General	Surrogate modeling	Fcn.	ACL	Maximin LHS	B-QBC, QB-MGP	Bayesian GP	[11]
Methods/General	Acceleration of BO	Fcn., Sim.	Batch BO	Sobol	EI	Vecchia GP Approximation	[78]
Methods/General	Surrogate modeling with reduction in computational complexity	Fcn.	ACL	LHS	Comb. of FI and ALC	GP	[151]
Methods/General	Surrogate modeling	Fcn., Sim.	ACL	LHS	ALC	Deep GP	[45]
Methods/General	Optimization of BO with function networks	Fcn., data sets	BO	-	KG	Function Network GP	[141]
Methods/General	Surrogate modeling	Fcn., Sim.	ACL	Random	ALM	HHK-GP	[13]
Methods/General	Extension of BO for specific target subsets	data sets	BO	Random	SwitchBAX, InfoBAX, MeanBAX	GP	[152]
Methods/General	Reduction in dimension with Sobol indices	Fcn., Exp.	ACL	LHS, Random	MUSIC	GP	[14]
Methods/General	Surrogate modeling of structural reliability analysis	Sim.	Batch ACL	LHS	qAK	GP	[146]
Methods/General	Surrogate modeling	Sim., Exp.	ACL	LHS	MSPE, IMSPE, ALC, ALM	Jump GP	[139]
Methods/General	Surrogate modeling with Sobol indices	Fcn.	ACL	Random	SBAL	GP	[103]

Table 4. Cont.

Application	Goal	Data	Type	Init.	Acq.	GP Model	References
Methods/General	Surrogate modeling	Fcn., Sim.	ACL	LHS	FI	PCEGP	[54]
Gaps	Transfer to industrial ADL loops is missing, scalable ALC and FI approximations are needed for large candidate sets, non-stationary cross-covariances in multi-output models are rarely used in practical studies						
Methods/Manufacturing	Optimization of power of free-electron laser	Fcn., data set	BO	Random	EI	Sparse online GPs	[57]
Method/Manufacturing	Optimization of BO for complex Fcn. networks	Fcn., Sim.	BO	Random	EI	GP network	[140]
Methods/Material science	Surrogate modeling of shape errors	data set	Batch ACL	Random	Diversity ALM	GP	[153]
Methods/Robotics	Surrogate modeling and safe exploration of different outputs	data set	ACL	Random	ALM	Multi-output GP	[133]
Methods/Robotics	Constrained and safe optimization	Exp.	BO	Predefined	SafeOpt	GP	[83]
Methods/Robotics	Robust optimization	Fcn., Sim.	BO	LHS	Robust EI	GP	[132]
Methods/Robotics	Surrogate modeling for robotic information gathering	Sim., Exp.	ACL	Pilot path	ALM	AKGP	[53]
Methods/Structural reliability	Surrogate modeling of structural reliability	Sim.	ACL	LHS	Distance-based	GP	[137]
Methods/Structural reliability	Surrogate modeling for structural reliability analysis	Sim.	Batch ACL	LHS	K-means prob. max.	GP	[138]
Gaps	Constraint-aware and safe acquisitions are reported but are not default choices in routine experimentation, and the industrial deployment of advanced non-stationary GP models remains limited; validation in real experimental loops is rare.						

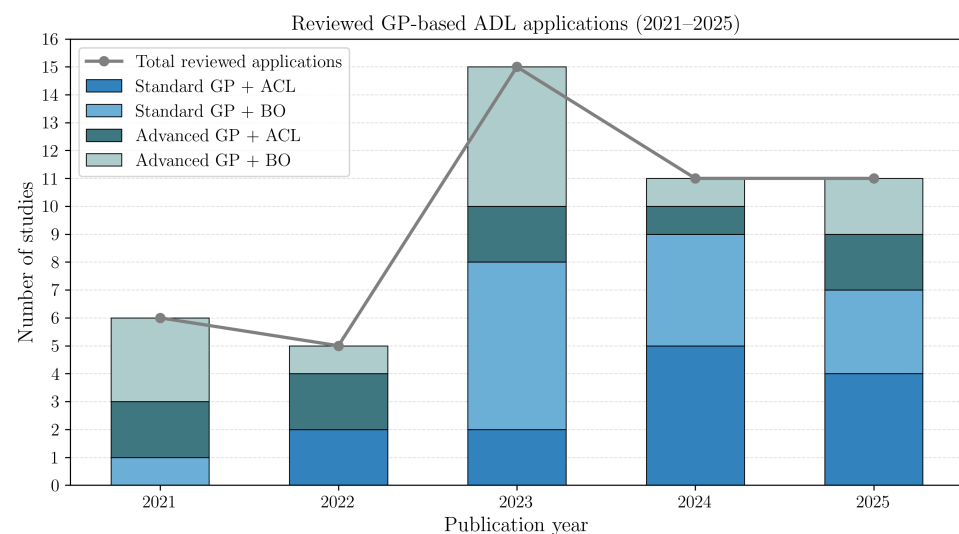


Figure 2. Trends in reviewed GP-based ADL applications

5. Software Libraries

In this section, an overview of widely used Python and R libraries for GP modeling, together with ACL and BO workflows, is provided. The intention is to indicate software defaults for practice and to point to reference implementations of advanced models. In Table 5, libraries by language and framework, together with a short characterization, as well as a citation, are summarized. The table is intended to guide practical selection, rather than to enumerate every feature.

5.1. Library Landscape

In this subsection, an overview of software stacks and model implementations for GP-based ACL and BO is provided, and the grouping and order follow the structure of Table 5.

5.1.1. Python Stacks for GP Modeling

Scalable GP regression is offered by GPyTorch, which provides fast linear algebra and convenient integration with BoTorch for BO as well with Pyro for fully Bayesian inference [111,154,155]. Scalable GP regression is also provided by HiGP [156], a recent Python package for large GP systems. Hierarchical kernel representations related to $\mathcal{H}^2$ -matrix ideas, Adaptive Factorized Nyström (AFN) preconditioning, and analytically derived gradients are used to improve training efficiency. In return, a different flexibility-performance trade-off is obtained than in general-purpose autodiff-based frameworks. For TensorFlow-based modeling, GPflow and GPflux are available and enable DGP constructions on top of GPflow [157,158]. Within scikit-learn, the GaussianProcessRegressor is provided as an accessible baseline and is integrated well with standard modeling pipelines [159]. Baseline GP modeling is also provided by GPy as a separate Python toolbox. Engineering-oriented use-cases can be supported with the Surrogate Modeling Toolbox (SMT), which is used for GP surrogates together with design of experiments utilities [160]. Multi-output regression can be prototyped with MOGPTK [161], and heteroscedastic regression in Python can be explored with hetGPy [162]. Fully Bayesian workflows are available via PyMC, which includes GP components and practical approximations for large covariance structures [163].

5.1.2. Optimization and Design Frameworks

Modular BO is provided by BoTorch and pairs naturally with GPyTorch for model training and Monte Carlo acquisitions [111]. Higher-level orchestration for online and offline experimentation is available in Ax, which builds on BoTorch and provides experiment management [164]. The TensorFlow stack is served by Trieste, which offers BO and ACL with support for constraints and multi-objective settings [165]. Backend-agnostic experiment design (i.e., independent of the underlying modeling library) that includes ACL, BO, and Bayesian quadrature is provided by Emukit [166].

Further options include scikit-optimize for sequential model-based optimization, Optuna for general-purpose hyperparameter optimization [167], modAL for a simple ACL API [168], and BatchBALD for information theoretic batch acquisitions [9]. Robust and scalable BO is addressed by Dragonfly [169], and research baselines are collected in RoBO.

5.1.3. Advanced Models

Reference implementations for advanced models, non-stationary kernels, local structure, and heteroscedastic noise are present in several entries. Non-stationary and local behavior can be explored with HHK-GP by [13], which serves as a reference implementation for advanced GP modeling in ACL studies. In the Python ecosystem, hetGPy provides a lightweight option for heteroscedastic regression, and MOGPTK supports multi-output modeling [161,162]. In the R ecosystem, non-stationary treed GPs are implemented in tgp [170], local approximate modeling at scale is provided by laGP [171], and heteroscedastic regression, together with sequential design utilities, is included in hetGP [172]. Deep hierarchical structures for uncertainty-aware surrogates are available via the deepgp package, which implements Bayesian deep GPs with examples for sequential design [45]. Kriging and BO with EGO and qEGO are operationalized in DiceKriging and DiceOptim [173], GPareto supports multi-objective BO [174], and ParBayesianOptimization and rBayesianOptimization provide convenient wrappers around GP-based BO in applied workflows.

5.1.4. R Ecosystem for GP, ACL, and BO

Long-standing and statistically grounded toolchains are available in the R ecosystem. Non-stationary treed GPs are implemented in `tgp` [170]. Local approximate modeling at scale is provided by `laGP` [171]. Heteroscedastic regression, together with sequential design utilities, is included in `hetGP` [172]. The `DiceKriging` and `DiceOptim` suite operationalizes Kriging, together with classical global optimization criteria such as EGO and its batch extension `qEGO` [173]. Multi-objective BO is supported by `GPareto` [174]. Parallel and user-friendly BO wrappers, which provide high-level interfaces around the underlying GP and optimization routines, are supplied by `ParBayesianOptimization` and `rBayesianOptimization`. Deep hierarchical GP models for sequential design can be constructed with the `deepgp` package [45].

5.2. Gaps and Guidance

In this subsection, observed gaps and maintenance aspects that are relevant for ADL deployments with GP models are first highlighted, and practical guidance for selecting software stacks and typical usage roles is then summarized.

5.2.1. Observed Gaps and Maintenance Notes

The coverage of multi-output ACL with non-stationary covariance structures is not reported in commonly used software libraries and requires custom extensions, which is consistent with gaps reported in the applications section. Integrated support for feasibility modeling and safe acquisitions is available in several BO frameworks. Workflows for heteroscedastic noise are well established in R packages and remain more restricted in Python, where tools such as `hetGPy` are mainly suited for prototyping [162,172]. These observations align with the deployment gaps identified in Table 4 and can guide practical library selection for ADL with GP models.

5.2.2. Guidance for Selection and Typical Roles

For small to medium data sets with GPU-based training, `GPYtorch` together with `BoTorch` provide convenient default tools [111,154]. For larger GP systems with an emphasis on numerical performance, `HiGP` can serve as an alternative Python option, especially when efficient kernel algebra and iterative solver performance are more important than broad autodiff-based modeling flexibility [156]. In environments centered around TensorFlow, `GPflow` and `Trieste` integrate well with that stack [157,165]. For baseline modeling and teaching, `scikit-learn` and `GPy` offer simple Gaussian process interfaces [159]. Multi-output regression can make use of `MOGPTK` [161] and `BoTorch` [111]. Heteroscedastic settings are supported by `hetGPy` in Python and `hetGP` in R [162,172]. The `hetGPy` package can be viewed as the Python-side counterpart to `hetGP`, although it currently serves more as a lightweight prototyping option, while the R package provides the more established workflow with sequential design utilities. In such heteroscedastic settings, initialization choices for the latent noise process, such as simple versus residual-based strategies, can affect the stability of the estimated noise structure and, therefore, the robustness of iterative ADL workflows. Engineering workflows that combine design of experiments and surrogate modeling can be organized with SMT and with `DiceKriging`/`DiceOptim` in R [160,173]. Information-theoretic or batch ACL studies can use `BatchBALD` and the batch utilities available in `BoTorch` and `Trieste` [9,111,165]. When non-stationarity is suspected, `HHK-GP`, `tgp`, `deepgp`, or `laGP` can be used [13,45,170,171].

Table 5. Overview of Python and R libraries for GP, ACL, and BO.

Library	Language/Framework	Characteristic strength	References
GPyTorch	Python/PyTorch	Scalable GP regression with fast linear algebra, smooth integration with BoTorch for BO and Pyro for full Bayesian modeling	[154]
HiGP	Python/Python with C++ backend	Scalable GP regression with hierarchical kernel representations, AFN-preconditioned iterative solvers, and analytically derived gradients	[156]
GPflow	Python/TensorFlow	Modular variational GP platform for research and applications	[157]
GPflux	Python/TensorFlow on GPflow	Deep GP constructions with variational building blocks interoperable with GPflow	[158]
GPy	Python/NumPy and SciPy	Classical toolbox with many kernels and an approachable API for education and baselines	
scikit-learn GaussianProcessRegressor	Python/NumPy and SciPy	Standard GP baselines integrated in scikit-learn pipelines, ARD options and common kernels such as RBF and Matérn	[159]
Pyro	Python/PyTorch	Probabilistic programming with GP priors and modern MCMC or variational inference for full Bayesian modeling	[155]
PyMC	Python/PyMC	Fully Bayesian modeling with GP modules and practical approximations such as HSGP	[163]
MOGPTK	Python/PyTorch	Multi-output GP toolkit with training utilities and diagnostics	[161]
hetGPy	Python/NumPy and SciPy	Lightweight prototyping for heteroscedastic GP regression in Python, Python-side counterpart to hetGP in R	[162]
SMT Surrogate Modeling Toolbox	Python/NumPy and SciPy	Engineering oriented surrogates with Kriging and GP, plus design of experiments utilities	[160]
HHK-GP	Python/GPflow	Non-stationary hyperplane kernel with ACL reference implementation	[13]
Gaps	Multi-output ACL with non-stationary covariance structures is not used		
BoTorch	Python/PyTorch	Modular BO with Monte Carlo acquisitions for research and production workflows	[111]
Ax Adaptive Experimentation Platform	Python/PyTorch stack	Orchestration for online and offline experimentation built on BoTorch	[164]
Trieste	Python/TensorFlow	BO and ACL on top of GPflow and GPflux with constraint and multi-objective support	[165]
Emukit	Python	Unified interface for experiment design, ACL, BO, and Bayesian quadrature	[166]
scikit-optimize	Python/sklearn ecosystem	Sequential model-based optimization with GP surrogates	
Optuna	Python	General-purpose hyperparameter optimization, complements GP-based BO stacks as a flexible HPO framework	[167]
modAL	Python/sklearn	Simple ACL API that works directly with sklearn estimators, including GP regressors	[168]
BatchBALD	Python/PyTorch	Information theoretic batch acquisitions for data efficient labeling and ACL	[9]
Dragonfly	Python	Robust and scalable BO including multi-fidelity and high-dimensional settings	[169]
RoBO Robust Bayesian Optimization	Python/sklearn	Research framework for robust BO baselines and benchmarks	
Gaps	Safe and feasible acquisitions are often available, scalable batch ACL utilities require custom integration		

Table 5. Cont.

Library	Language/Framework	Characteristic strength	References
tgp	R	Non-stationary treed GP modeling with ACL	[170]
laGP	R	Local approximate GPs for large data	[171]
hetGP	R	Heteroscedastic GP regression with sequential design criteria such as IMSPE, established R counterpart to hetGPy in Python	[172]
DiceKriging and DiceOptim	R	Kriging and BO with EGO and qEGO criteria widely used in engineering design	[173]
GPareto	R	Multi-objective BO with GP surrogates and Pareto analysis	[174]
deepgp	R	Bayesian deep GP modeling with examples for sequential design	[45]
ParBayesianOptimization	R	Parallel BO wrappers often used with GP-based surrogates	
rBayesianOptimization	R	Lightweight BO interface for applied workflows based on GP surrogates	
Gaps	Heteroscedastic regression and sequential design are more standardized in R, multi-output ACL with non-stationary covariance structures is not used		

6. Summary and Outlook

A consolidated overview of GP models is provided, together with their use in ADL. Stationary baselines and advanced classes, including non-stationary, heteroscedastic, sparse, local, dynamic, and multi-output variants, are summarized. Both ACL and BO for GPs are reviewed with commonly used acquisition functions, and the applications literature is mapped by domain, initialization, acquisition, and model choice with an emphasis on the last five years and a small number of important earlier works.

Consistent patterns across domains are observed. In manufacturing, the standard GP is used almost exclusively in both simulation and production studies because training is reliable, calibration is stable under small data, and maintenance risk is low. In chemical engineering, BO with space-filling starts is common for experimental applications, while multi-objective settings motivate multi-output modeling when several quality metrics have to be optimized jointly. In structural reliability, heteroscedastic likelihoods are frequently required due to input-dependent measurement variance, and ACL with global variance reduction is standard after an initial space-filling design. In aerospace and other simulation-heavy fields, non-stationary and local models are used more often to represent heterogeneous physics and regime changes, although deployment in industrial ADL loops remains rare. In robotics, constraint-aware and safe acquisitions are reported together with robust objectives. Across domains, batch ACL for global surrogates is not present. Multi-output ACL with non-stationary covariance structures is also an open point for simulations and practical applications, although many applications involve coupled responses with input-dependent behavior.

The contribution of this review is to deliver practical guidance grounded in a structured synthesis of methods and applications. A mapping from goal and data context to model and acquisition is provided. Situations are identified where the standard GP with simple acquisitions is sufficient, and cases are highlighted where multi-output structure, heteroscedastic likelihoods, or non-stationary kernels are justified. Typical deployment issues are made explicit, along with potential solutions. The applications table and the software-library overview reduce setup time and support tool choice under constraints such as batching, multi-objective goals, ACL workflows, heteroscedastic noise, and non-stationary behavior.

An outlook focused on closing observed gaps is offered. Explainable ADL loops should also be developed more explicitly, so that model selection, residual-based upgrades, acquisition decisions, and uncertainty contributions remain interpretable in practical deployment. The use of non-stationary modeling in real ADL loops would benefit from identifiable parameterizations, informative priors, and residual-driven upgrades that are introduced only when diagnostic evidence indicates that standard stationary models fail. Robust likelihoods that handle outliers should be used systematically outside structural reliability, in particular in manufacturing and chemistry, where measurement noise and process drift are common. Constraint-aware and safe acquisitions should become default elements of experimental applications. When several objectives or sensors are coupled, multi-output acquisitions and non-stationary cross-covariances are needed to reflect the true decision space. Batch ACL for global surrogates should be paired with scalable approximations to ALC and FI so that credible uncertainty is preserved for large candidate sets. Posterior multi-modality should be addressed explicitly through tempered transitions to stabilize selection under low data. The transfer from simulation to experiment should be supported more explicitly through multi-fidelity GP frameworks and transfer learning strategies that combine abundant lower-cost simulations with a limited number of high-fidelity experiments.

For practitioners, a route from application context to reliable defaults and targeted upgrades is outlined. For researchers, a set of priorities is suggested that may lower barriers to deploying advanced GP models within ADL loops and that can support credible uncertainty, efficient data use, and robust decisions across simulations, experiments, and production systems.

Author Contributions: The contributions of the authors to this work are as follows: literature review, D.P.; investigation, D.P.; writing—original draft preparation, D.P.; writing—review and editing, D.P., E.A. and D.S.; visualization, D.P.; supervision, E.A. and D.S.; project administration, E.A. and D.S.; final proofreading, D.S. All authors have read and agreed to the published version of the manuscript.

Funding: This work has been partially sponsored by the German Federal Ministry of Education and Research in the funding program “Forschung an Fachhochschulen”, project I2DACH (grant no. 13FH557KX0, <https://www.hs-niederrhein.de/i2dach>, accessed on 2 April 2026).

Data Availability Statement: No data was used for the research described in the article.

Acknowledgments: The authors acknowledge the use of ChatGPT (version 5, <https://chat.openai.com>) by OpenAI, for initial text refinement and stylistic improvements. The tool was employed to save time and partly enhance the quality of the writing. All generated outputs were subsequently reviewed and corrected to ensure accuracy and alignment with the intended statements. Furthermore, the authors acknowledge the use of DeepL (<https://www.deepl.com>) in optimizing the phrasing and clarity of the text, particularly in refining the English formulations of technical content. All translations and suggestions provided by DeepL were carefully reviewed and, where necessary, adjusted to ensure correctness and fidelity to the original meaning. The authors acknowledge support from the Open Access Publication Fund of the University of Duisburg-Essen.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Di Fiore, F.; Nardelli, M.; Mainini, L. Active Learning and Bayesian Optimization: A Unified Perspective to Learn with a Goal. *Arch. Comput. Methods Eng.* **2024**, *31*, 2985–3013. [\[CrossRef\]](#)
2. Greenhill, S.; Rana, S.; Gupta, S.; Vellanki, P.; Venkatesh, S. Bayesian Optimization for Adaptive Experimental Design: A Review. *IEEE Access* **2020**, *8*, 13937–13948. [\[CrossRef\]](#)
3. Gramacy, R.B. *Surrogates: Gaussian Process Modeling, Design and Optimization for the Applied Sciences*; Chapman Hall/CRC: Boca Raton, FL, USA, 2020.

4. Shahriari, B.; Swersky, K.; Wang, Z.; Adams, R.P.; de Freitas, N. Taking the Human Out of the Loop: A Review of Bayesian Optimization. *Proc. IEEE* **2016**, *104*, 148–175. [[CrossRef](#)]
5. Daulton, S.; Eriksson, D.; Balandat, M.; Bakshy, E. Multi-objective Bayesian optimization over high-dimensional search spaces. In *Proceedings of the Thirty-Eighth Conference on Uncertainty in Artificial Intelligence, Eindhoven, The Netherlands, 1–5 August 2022*; Proceedings of Machine Learning Research; PMLR: Cambridge, MA, USA, 2022; Volume 180, pp. 507–517.
6. Wang, K.; Dowling, A.W. Bayesian Optimization for Chemical Products and Functional Materials. *Curr. Opin. Chem. Eng.* **2022**, *36*, 100728. [[CrossRef](#)]
7. Turner, R.; Eriksson, D.; McCourt, M.; Kiili, J.; Laaksonen, E.; Xu, Z.; Guyon, I. Bayesian Optimization is Superior to Random Search for Machine Learning Hyperparameter Tuning: Analysis of the Black-Box Optimization Challenge 2020. In *Proceedings of the NeurIPS 2020 Competition and Demonstration Track*; Proceedings of Machine Learning Research; PMLR: Cambridge, MA, USA, 2021; Volume 133, pp. 3–26.
8. Settles, B. *Active Learning*; Springer International Publishing: Cham, Switzerland, 2012.
9. Kirsch, A.; van Amersfoort, J.; Gal, Y. BatchBALD: Efficient and Diverse Batch Acquisition for Deep Bayesian Active Learning. In *Proceedings of the Advances in Neural Information Processing Systems*; Curran Associates, Inc.: Red Hook, NY, USA, 2019; Volume 32, pp. 7026–7037.
10. Rasmussen, C.; Williams, C. *Gaussian Process for Machine Learning*; The MIT Press: London, UK, 2006.
11. Riis, C.; Antunes, F.; Hüttel, F.; Lima Azevedo, C.; Pereira, F. Bayesian Active Learning with Fully Bayesian Gaussian Processes. In *Proceedings of the Advances in Neural Information Processing Systems*; Curran Associates, Inc.: Red Hook, NY, USA, 2022; Volume 35, pp. 12141–12153.
12. Marmin, S.; Ginsbourger, D.; Baccou, J.; Liandrat, J. Warped Gaussian Processes and Derivative-Based Sequential Designs for Functions with Heterogeneous Variations. *SIAM/ASA J. Uncertain. Quantif.* **2018**, *6*, 991–1018. [[CrossRef](#)]
13. Bitzer, M.; Meister, M.; Zimmer, C. Hierarchical-Hyperplane Kernels for Actively Learning Gaussian Process Models of Nonstationary Systems. In *Proceedings of the 26th International Conference on Artificial Intelligence and Statistics, Valencia, Spain, 25–27 April 2023*; Proceedings of Machine Learning Research; PMLR: Cambridge, MA, USA, 2023; Volume 206, pp. 7897–7912.
14. Chauhan, M.S.; Ojeda-Tuz, M.; Catarelli, R.A.; Gurley, K.R.; Tsapetis, D.; Shields, M.D. On Active Learning for Gaussian Process-based Global Sensitivity Analysis. *Reliab. Eng. Syst. Saf.* **2024**, *245*, 109945. [[CrossRef](#)]
15. Yue, X.; Wen, Y.; Hunt, J.H.; Shi, J. Active Learning for Gaussian Process Considering Uncertainties with Application to Shape Control of Composite Fuselage. *IEEE Trans. Autom. Sci. Eng.* **2021**, *18*, 36–46. [[CrossRef](#)]
16. Booth, A.S.; Cooper, A.; Gramacy, R.B. Nonstationary Gaussian Process Surrogates. *arXiv* **2023**, arXiv:2305.19242. [[CrossRef](#)]
17. Jakkala, K. Deep Gaussian Processes: A Survey. *arXiv* **2021**, arXiv:2106.12135. [[CrossRef](#)]
18. Li, P.; Chen, S. A review on Gaussian Process Latent Variable Models. *CAAI Trans. Intell. Technol.* **2016**, *1*, 366–376. [[CrossRef](#)]
19. Liu, H.; Ong, Y.S.; Shen, X.; Cai, J. When Gaussian Process Meets Big Data: A Review of Scalable GPs. *IEEE Trans. Neural Netw. Learn. Syst.* **2020**, *31*, 4405–4423. [[CrossRef](#)] [[PubMed](#)]
20. Lyu, C.; Liu, X.; Mihaylova, L. Review of Recent Advances in Gaussian Process Regression Methods. In *Advances in Computational Intelligence Systems*; Advances in Intelligent Systems and Computing; Springer Nature: Cham, Switzerland, 2024; Volume 1454, pp. 226–237.
21. Marrel, A.; Iooss, B. Probabilistic Surrogate Modeling by Gaussian Process: A Review on Recent Insights in Estimation and Validation. *Reliab. Eng. Syst. Saf.* **2024**, *247*, 110094. [[CrossRef](#)]
22. Swiler, L.P.; Gulian, M.; Frankel, A.L.; Safta, C.; Jakeman, J.D. A Survey of Constrained Gaussian Process Regression: Approaches and Implementation Challenges. *J. Mach. Learn. Model. Comput.* **2020**, *1*, 119–156. [[CrossRef](#)]
23. Scampicchio, A.; Arcari, E.; Lahr, A.; Zeilinger, M.N. Gaussian Processes for Dynamics Learning in Model Predictive Control. *Annu. Rev. Control* **2025**, *60*, 101034. [[CrossRef](#)]
24. Binois, M.; Wycoff, N. A Survey on High-dimensional Gaussian Process Modeling with Application to Bayesian Optimization. *ACM Trans. Evol. Learn. Optim.* **2022**, *2*, 1–26. [[CrossRef](#)]
25. Kumar, P.; Gupta, A. Active Learning Query Strategies for Classification, Regression, and Clustering: A Survey. *J. Comput. Sci. Technol.* **2020**, *35*, 913–945. [[CrossRef](#)]
26. Malu, M.; Dasarathy, G.; Spanias, A. Bayesian Optimization in High-Dimensional Spaces: A Brief Survey. In *Proceedings of the 2021 12th International Conference on Information, Intelligence, Systems & Applications (IISA)*, Chania Crete, Greece, 12–14 July 2021; pp. 1–8.
27. Kochan, D.; Yang, X. Gaussian Process Regression with Soft Equality Constraints. *Mathematics* **2025**, *13*, 353. [[CrossRef](#)]
28. Polke, D.; Kösters, T.; Ahle, E.; Söffker, D. Polynomial Chaos Expanded Gaussian Process. *Mach. Learn. Knowl. Extr. (MAKE)* **2026**, *8*, 78. [[CrossRef](#)]
29. Kocijan, J. *Modelling and Control of Dynamic Systems Using Gaussian Process Models*; Advances in Industrial Control; Springer: Cham, Switzerland, 2016.

30. Moreno-Muñoz, P.; Artés, A.; Álvarez, M. Heterogeneous Multi-output Gaussian Process Prediction. In *Proceedings of the Advances in Neural Information Processing Systems, Montréal, Canada, 3–8 December 2018*; Curran Associates, Inc.: Red Hook, NY, USA, 2018; Volume 31, pp. 6711–6720.
31. Sauer, A.; Cooper, A.; Gramacy, R.B. Vecchia-Approximated Deep Gaussian Processes for Computer Experiments. *J. Comput. Graph. Stat.* **2023**, *32*, 824–837. [\[CrossRef\]](#)
32. Li, J.; Pan, L.; Suvama, M.; Wang, X. Machine Learning aided Supercritical Water Gasification for H₂-rich Syngas Production with Process Optimization and Catalyst Screening. *Chem. Eng. J.* **2021**, *426*, 131285. [\[CrossRef\]](#)
33. Kiran, B.R.; Sobh, I.; Talpaert, V.; Mannion, P.; Sallab, A.; Yogamani, S.; Pérez, P. Deep Reinforcement Learning for Autonomous Driving: A Survey. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 4909–4926. [\[CrossRef\]](#)
34. Somu, N.; Raman, G.; Ramamritham, K. A Deep Learning Framework for Building Energy Consumption Forecast. *Renew. Sustain. Energy Rev.* **2021**, *137*, 110591. [\[CrossRef\]](#)
35. Keogh, E.; Mueen, A. Curse of Dimensionality. In *Encyclopedia of Machine Learning*; Springer: Boston, MA, USA, 2011; pp. 257–258.
36. Neal, R.M. *Bayesian Learning for Neural Networks*; Lecture Notes in Statistics; Springer: New York, NY, USA, 1996; Volume 118.
37. Paananen, T.; Piiroinen, J.; Andersen, M.R.; Vehtari, A. Variable Selection for Gaussian Processes via Sensitivity Analysis of the Posterior Predictive Distribution. In *Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics, Naha, Okinawa, Japan, 16–18 April 2019*; Proceedings of Machine Learning Research; PMLR: Cambridge, MA, USA, 2019; Volume 89, pp. 1743–1752.
38. Varunram, T.N.; Shivaprasad, M.B.; Aishwarya, K.H.; Balraj, A.; Savish, S.V.; Ullas, S. Analysis of Different Dimensionality Reduction Techniques and Machine Learning Algorithms for an Intrusion Detection System. In *Proceedings of the 2021 IEEE 6th International Conference on Computing, Communication and Automation (ICCCA), Arad, Romania, 17–19 December 2021*; pp. 237–242.
39. Paciorek, C.; Schervish, M. Nonstationary Covariance Functions for Gaussian Process Regression. In *Proceedings of the Advances in Neural Information Processing Systems*; MIT Press: Cambridge, MA, USA, 2003; Volume 16, pp. 273–280.
40. Plagemann, C.; Kersting, K.; Burgard, W. Nonstationary Gaussian Process Regression Using Point Estimates of Local Smoothness. In *Proceedings of the Machine Learning and Knowledge Discovery in Databases*; Springer: Berlin/Heidelberg, Germany, 2008; pp. 204–219.
41. Gramacy, R.B.; Lee, H.K.H. Bayesian Treed Gaussian Process Models with an Application to Computer Modeling. *J. Am. Stat. Assoc.* **2008**, *103*, 1119–1130. [\[CrossRef\]](#)
42. Gramacy, R.B.; Lee, H.K.H.; MacReady, W. Parameter Space Exploration with Gaussian Process Trees. In *Proceedings of the Twenty-First International Conference on Machine Learning, Banff, Alberta, Canada, 4–8 July 2004*; Omnipress: Madison, WI, USA, 2004; pp. 353–360.
43. Wilson, A.G.; Knowles, D.A.; Ghahramani, Z. Gaussian Process Regression Networks. In *Proceedings of the 29th International Conference on International Conference on Machine Learning, Madison, WI, USA, 26 June–1 July 2012*; pp. 1139–1146.
44. Damianou, A.; Lawrence, N.D. Deep Gaussian Processes. In *Proceedings of the Sixteenth International Conference on Artificial Intelligence and Statistics, Scottsdale, AZ, USA, 29 April–1 May 2013*; Proceedings of Machine Learning Research; PMLR: Cambridge, MA, USA, 2013; Volume 31, pp. 207–215.
45. Sauer, A.; Gramacy, R.B.; Higdon, D. Active Learning for Deep Gaussian Process Surrogates. *Technometrics* **2023**, *65*, 4–18. [\[CrossRef\]](#)
46. Wilson, A.G.; Hu, Z.; Salakhutdinov, R.; Xing, E.P. Deep Kernel Learning. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics, Cadiz, Spain, 9–11 May 2016*; Proceedings of Machine Learning Research; PMLR: Cambridge, MA, USA, 2016; Volume 51, pp. 370–378.
47. Heinonen, M.; Mannerström, H.; Rousu, J.; Kaski, S.; Lähdesmäki, H. Non-Stationary Gaussian Process Regression with Hamiltonian Monte Carlo. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics, Cadiz, Spain, 9–11 May 2016*; Proceedings of Machine Learning Research; PMLR: Cambridge, MA, USA, 2016; Volume 51, pp. 732–740.
48. Remes, S.; Heinonen, M.; Kaski, S. Non-Stationary Spectral Kernels. In *Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017*; Curran Associates, Inc.: Red Hook, NY, USA, 2017; Volume 30, pp. 4642–4651.
49. Cremanns, K.; Roos, D. Deep Gaussian Covariance Network. *arXiv* **2017**, arXiv:1710.06202. [\[CrossRef\]](#)
50. Liang, Y.; Li, S.; Yan, C.; Li, M.; Jiang, C. Explaining the Black-box Model: A Survey of Local Interpretation Methods for Deep Neural Networks. *Neurocomputing* **2021**, *419*, 168–182. [\[CrossRef\]](#)
51. Park, C. Jump Gaussian Process Model for Estimating Piecewise Continuous Regression Functions. *J. Mach. Learn. Res.* **2022**, *23*, 1–37.
52. Xu, Y.; Park, C. Deep Jump Gaussian Processes for Surrogate Modeling of High-Dimensional Piecewise Continuous Functions. *arXiv* **2026**, arXiv:2510.21974.
53. Chen, W.; Khardon, R.; Liu, L. Adaptive Robotic Information Gathering via Non-stationary Gaussian Processes. *Int. J. Robot. Res.* **2024**, *43*, 405–436. [\[CrossRef\]](#)

54. Polke, D.; Ahle, E.; Söffker, D. Bayesian Active Learning with Polynomial Chaos Expanded Gaussian Process. In Proceedings of the 2026 IEEE 3rd International Conference on Artificial Intelligence, Computer, Data Sciences and Applications (ACDSA), Boracay Island, Philippines, 5–7 February 2026; pp. 1–14.
55. Hensman, J.; Matthews, A.G.d.G.; Filippone, M.; Ghahramani, Z. MCMC for Variationally Sparse Gaussian Processes. In Proceedings of the 29th International Conference on Neural Information Processing Systems, Montreal, QC, Canada, 7–12 December 2015; Volume 1, pp. 1648–1656.
56. Bauer, M.; van der Wilk, M.; Rasmussen, C.E. Understanding probabilistic sparse Gaussian process approximations. In *Proceedings of the 30th International Conference on Neural Information Processing Systems, Barcelona, Spain*; Curran Associates, Inc.: Red Hook, NY, USA, 2016; pp. 1533–1541.
57. McIntire, M.; Ratner, D.; Ermon, S. Sparse Gaussian processes for Bayesian optimization. In *Proceedings of the Thirty-Second Conference on Uncertainty in Artificial Intelligence, Jersey City, New Jersey, USA*; AUAI Press: Arlington, VA, USA, 2016; pp. 517–526.
58. Almosallam, I.A.; Jarvis, M.J.; Roberts, S.J. GPz: Non-stationary Sparse Gaussian Processes for Heteroscedastic Uncertainty Estimation in Photometric Redshifts. *Mon. Not. R. Astron. Soc.* **2016**, *462*, 726–739. [[CrossRef](#)]
59. Cheng, L.F.; Dumitrascu, B.; Darnell, G.; Chivers, C.; Draugelis, M.; Li, K.; Engelhardt, B.E. Sparse Multi-output Gaussian Processes for Online Medical Time Series Prediction. *BMC Med. Inform. Decis. Mak.* **2020**, *20*, 152. [[CrossRef](#)]
60. Luo, H.; Nattino, G.; Pratola, M.T. Sparse Additive Gaussian Process Regression. *J. Mach. Learn. Res.* **2022**, *23*, 1–34.
61. Hajibabaei, A.; Myung, C.W.; Kim, K.S. Sparse Gaussian Process Potentials: Application to Lithium Diffusivity in Superionic Conducting Solid Electrolytes. *Phys. Rev. B* **2021**, *103*, 214102. [[CrossRef](#)]
62. Yang, K.; Lu, J.; Wan, W.; Zhang, G.; Hou, L. Transfer Learning Based on Sparse Gaussian Process for Regression. *Inf. Sci.* **2022**, *605*, 286–300. [[CrossRef](#)]
63. Hewing, L.; Kabzan, J.; Zeilinger, M.N. Cautious Model Predictive Control using Gaussian Process Regression. *IEEE Trans. Control Syst. Technol.* **2020**, *28*, 2736–2743. [[CrossRef](#)]
64. Diepers, F.; Polke, D.; Ahle, E.; Söffker, D. Comparison of Different Gaussian Process Models and Applications in Model Predictive Control. In Proceedings of the 2023 23rd International Conference on Control, Automation and Systems (ICCAS), Yeosu, Republic of Korea, 17–20 October 2023; pp. 54–59.
65. Diepers, F.; Ahle, E.; Söffker, D. Investigation of the Influence of Training Data and Methods on the Control Performance of MPC Utilizing Gaussian Processes. In *Systems Theory in Data and Optimization; Lecture Notes in Control and Information Sciences—Proceedings*; Springer Nature: Cham, Switzerland, 2025; pp. 87–103.
66. Chakraborty, S.; Adhikari, S.; Ganguli, R. The Role of Surrogate Models in the Development of Digital Twins of Dynamic Systems. *Appl. Math. Model.* **2021**, *90*, 662–681. [[CrossRef](#)]
67. Bilonis, I.; Zabarar, N. Multi-output Local Gaussian Process Regression: Applications to Uncertainty Quantification. *J. Comput. Phys.* **2012**, *231*, 5718–5746. [[CrossRef](#)]
68. Alaa, A.M.; van der Schaar, M. Bayesian Inference of Individualized Treatment Effects using Multi-task Gaussian Processes. In *Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017*; Guyon, I., Von Luxburg, U., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2017; Volume 30, pp. 3427–3435.
69. Parra, G.; Tobar, F. Spectral Mixture Kernels for Multi-Output Gaussian Processes. In *Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017*; Curran Associates, Inc.: Red Hook, NY, USA, 2017; Volume 30, pp. 6681–6690.
70. Liu, H.; Cai, J.; Ong, Y.S. Remarks on Multi-output Gaussian Process Regression. *Knowl.-Based Syst.* **2018**, *144*, 102–121. [[CrossRef](#)]
71. Mehta, M.; Shao, C. Adaptive Sampling Design for Multi-task Learning of Gaussian Processes in Manufacturing. *J. Manuf. Syst.* **2021**, *61*, 326–337. [[CrossRef](#)]
72. Gramacy, R.B.; Apley, D.W. Local Gaussian Process Approximation for Large Computer Experiments. *J. Comput. Graph. Stat.* **2015**, *24*, 561–578. [[CrossRef](#)]
73. Park, C.; Huang, J.Z. Efficient Computation of Gaussian Process Regression for Large Spatial Data Sets by Patching Local Gaussian Processes. *J. Mach. Learn. Res.* **2016**, *17*, 1–29.
74. Fuhg, J.N.; Marino, M.; Bouklas, N. Local Approximate Gaussian Process Regression for Data-driven Constitutive Models: Development and Comparison with Neural Networks. *Comput. Methods Appl. Mech. Eng.* **2022**, *388*, 114217. [[CrossRef](#)]
75. Liu, C.; Duan, Z.; Zhang, B.; Zhao, Y.; Yuan, Z.; Zhang, Y.; Wu, Y.; Jiang, Y.; Tai, H. Local Gaussian Process Regression with Small Sample Data for Temperature and Humidity Compensation of Polyaniline-cerium Dioxide NH₃ Sensor. *Sens. Actuators B Chem.* **2023**, *378*, 133113. [[CrossRef](#)]
76. Katzfuss, M.; Guinness, J.; Gong, W.; Zilber, D. Vecchia Approximations of Gaussian-Process Predictions. *J. Agric. Biol. Environ. Stat.* **2020**, *25*, 383–414. [[CrossRef](#)]
77. Katzfuss, M.; Guinness, J. A General Framework for Vecchia Approximations of Gaussian Processes. *Stat. Sci.* **2021**, *36*, 124–141. [[CrossRef](#)]

78. Jimenez, F.; Katzfuss, M. Scalable Bayesian Optimization using Vecchia Approximations of Gaussian Processes. In *Proceedings of the 26th International Conference on Artificial Intelligence and Statistics, Valencia, Spain, 25–27 April 2023*; Proceedings of Machine Learning Research; PMLR: Cambridge, MA, USA, 2023; Volume 206, pp. 1492–1512.
79. Pleiss, G.; Gardner, J.; Weinberger, K.; Wilson, A.G. Constant-Time Predictive Distributions for Gaussian Processes. In *Proceedings of the 35th International Conference on Machine Learning, Stockholm, Sweden, 10–15 July 2018*; Proceedings of Machine Learning Research; PMLR: Cambridge, MA, USA, 2018; Volume 80, pp. 4114–4123.
80. Schöbi, R.; Sudret, B.; Wiart, J. Polynomial-Chaos-based Kriging. *Int. J. Uncertain. Quantif.* **2015**, *5*, 171–193. [[CrossRef](#)]
81. Sigrist, F. Gaussian Process Boosting. *J. Mach. Learn. Res.* **2022**, *23*, 1–46.
82. Victoria, A.H.; Maragatham, G. Automatic Tuning of Hyperparameters Using Bayesian Optimization. *Evol. Syst.* **2021**, *12*, 217–223. [[CrossRef](#)]
83. Berkenkamp, F.; Krause, A.; Schoellig, A.P. Bayesian Optimization with Safety Constraints: Safe and Automatic Parameter Tuning in Robotics. *Mach. Learn.* **2023**, *112*, 3713–3747. [[CrossRef](#)]
84. McDonald, M.A.; Koscher, B.A.; Canty, R.B.; Zhang, J.; Ning, A.; Jensen, K.F. Bayesian Optimization over Multiple Experimental Fidelities Accelerates Automated Discovery of Drug Molecules. *ACS Cent. Sci.* **2025**, *11*, 346–356. [[CrossRef](#)]
85. Peralta, F.; Reina, D.G.; Toral, S.; Arzamendia, M.; Gregor, D. A Bayesian Optimization Approach for Multi-Function Estimation for Environmental Monitoring Using an Autonomous Surface Vehicle: Ypacarai Lake Case Study. *Electronics* **2021**, *10*, 963. [[CrossRef](#)]
86. Golestan, S.; Ardakanian, O.; Boulanger, P. Grey-Box Bayesian Optimization for Sensor Placement in Assisted Living Environments. In *Proceedings of the AAAI Conference on Artificial Intelligence, Vancouver, BC, Canada, 20–27 February 2024*; Volume 38, pp. 22049–22057.
87. Deneault, J.R.; Chang, J.; Myung, J.; Hooper, D.; Armstrong, A.; Pitt, M.; Maruyama, B. Toward Autonomous Additive Manufacturing: Bayesian Optimization on a 3D Printer. *MRS Bull.* **2021**, *46*, 566–575. [[CrossRef](#)]
88. Polke, D.; Surjana, A.; Diepers, F.; Ahle, E.; Söffker, D. Development of a Modular Automation Framework for Data-Driven Modeling and Optimization of Coating Formulations. In *Proceedings of the 2023 IEEE 28th International Conference on Emerging Technologies and Factory Automation (ETFA), Sinaia, Romania, 12–15 September 2023*; pp. 1–8.
89. Lookman, T.; Balachandran, P.V.; Xue, D.; Yuan, R. Active Learning in Materials Science with Emphasis on Adaptive Sampling Using Uncertainties for Targeted Design. *npj Comput. Mater.* **2019**, *5*, 21. [[CrossRef](#)]
90. Pronzato, L.; Müller, W.G. Design of Computer Experiments: Space Filling and Beyond. *Stat. Comput.* **2012**, *22*, 681–701. [[CrossRef](#)]
91. Renardy, M.; Joslyn, L.R.; Millar, J.A.; Kirschner, D.E. To Sobol or not to Sobol? The effects of sampling schemes in systems biology applications. *Math. Biosci.* **2021**, *337*, 108593. [[CrossRef](#)]
92. McKay, M.D.; Beckman, R.J.; Conover, W.J. A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code. *Technometrics* **1979**, *21*, 239–245. [[PubMed](#)]
93. Lin, C.D.; Tang, B. Latin Hypercubes and Space-filling Designs. In *Handbook of Design and Analysis of Experiments*; Dean, A., Morris, M., Stufken, J., Bingham, D., Eds.; Chapman & Hall/CRC Press: Boca Raton, FL, USA, 2015; pp. 593–625.
94. Keller, A.; Van Keirsbilck, M. Artificial Neural Networks Generated by Low Discrepancy Sequences. In *Monte Carlo and Quasi-Monte Carlo Methods*; Springer Proceedings in Mathematics & Statistics; Springer International Publishing: Cham, Switzerland, 2022; Volume 387, pp. 291–311.
95. Johnson, M.; Moore, L.; Ylvisaker, D. Minimax and Maximin Distance Designs. *J. Stat. Plan. Inference* **1990**, *26*, 131–148. [[CrossRef](#)]
96. Zhang, B.; Cole, A.D.; Gramacy, R.B. Distance-Distributed Design for Gaussian Process Surrogates. *Technometrics* **2021**, *63*, 40–52. [[CrossRef](#)]
97. Azriel, D. Optimal Minimax Random Designs for Weighted Least Squares Estimators. *Biometrika* **2022**, *110*, 273–280. [[CrossRef](#)]
98. Jankovic, A.; Chaudhary, G.; Goia, F. Designing the Design of Experiments (DOE)—An Investigation on the Influence of Different Factorial Designs on the Characterization of Complex Systems. *Energy Build.* **2021**, *250*, 111298. [[CrossRef](#)]
99. MacKay, D.J.C. Information-Based Objective Functions for Active Data Selection. *Neural Comput.* **1992**, *4*, 590–604. [[CrossRef](#)]
100. Cohn, D.; Ghahramani, Z.; Jordan, M. Active Learning with Statistical Models. In *Proceedings of the Advances in Neural Information Processing Systems*; MIT Press: Cambridge, MA, USA, 1994; Volume 7.
101. Seo, S.; Wallat, M.; Graepel, T.; Obermayer, K. Gaussian Process Regression: Active Data Selection and Test Point Rejection. In *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks. IJCNN 2000. Neural Computing: New Challenges and Perspectives for the New Millennium, Como, Italy, 24–27 July 2000*; pp. 241–246.
102. Sahoo, R.N.; Gakhar, S.; Rejith, R.G.; Verrelst, J.; Ranjan, R.; Kondraju, T.; Meena, M.C.; Mukherjee, J.; Daas, A.; Kumar, S.; et al. Optimizing the Retrieval of Wheat Crop Traits from UAV-Borne Hyperspectral Image with Radiative Transfer Modelling Using Gaussian Process Regression. *Remote Sens.* **2023**, *15*, 5496. [[CrossRef](#)]

103. Wulf, B.; Polke, D.; Ahle, E. Active Learning for Gaussian Processes Based on Global Sensitivity Analysis. In Proceedings of the 2025 IEEE 4th International Conference on Computing and Machine Intelligence (ICMI), Mount Pleasant, MI, USA, 5–6 April 2025; pp. 1–7.
104. Sobol, I. Global Sensitivity Indices for Nonlinear Mathematical Models and Their Monte Carlo Estimates. *Math. Comput. Simul.* **2001**, *55*, 271–280. [\[CrossRef\]](#)
105. Schmitz, C.; Cremanns, K.; Bissadi, G. Application of Machine Learning Algorithms for Use in Material Chemistry. In *Computational and Data-Driven Chemistry Using Artificial Intelligence*; Elsevier: Amsterdam, The Netherlands, 2022; pp. 161–192.
106. Wu, J.; Chen, X.Y.; Zhang, H.; Xiong, L.D.; Lei, H.; Deng, S.H. Hyperparameter Optimization for Machine Learning Models Based on Bayesian Optimization. *J. Electron. Sci. Technol.* **2019**, *17*, 26–40.
107. Nguyen, V. Bayesian Optimization for Accelerating Hyper-Parameter Tuning. In Proceedings of the 2019 IEEE Second International Conference on Artificial Intelligence and Knowledge Engineering (AIKE), Sardinia, Italy, 3–5 June 2019; pp. 302–305.
108. Ozaki, Y.; Tanigaki, Y.; Watanabe, S.; Onishi, M. Multiobjective Tree-structured Parzen Estimator for Computationally Expensive Optimization Problems. In Proceedings of the 2020 Genetic and Evolutionary Computation Conference, Cancún, Mexico, 8–12 July 2020; pp. 533–541.
109. Liu, D.C.; Nocedal, J. On the Limited Memory BFGS Method for Large Scale Optimization. *Math. Program.* **1989**, *45*, 503–528. [\[CrossRef\]](#)
110. Dozat, T. Incorporating Nesterov Momentum into Adam. In Proceedings of the 4th International Conference on Learning Representations, Workshop Track, San Juan, Puerto Rico, 2–4 May 2016; pp. 1–4.
111. Balandat, M.; Karrer, B.; Jiang, D.; Daulton, S.; Letham, B.; Wilson, A.G.; Bakshy, E. BoTorch: A Framework for Efficient Monte-Carlo Bayesian Optimization. In *Proceedings of the 34th International Conference on Neural Information Processing Systems, Vancouver, BC, Canada*; Curran Associates, Inc.: Red Hook, NY, USA, 2020; pp. 21524–21538.
112. Page, M.J.; McKenzie, J.E.; Bossuyt, P.M.; Boutron, I.; Hoffmann, T.C.; Mulrow, C.D.; Shamseer, L.; Tetzlaff, J.M.; Akl, E.A.; Brennan, S.E.; et al. The PRISMA 2020 Statement: An Updated Guideline for Reporting Systematic Reviews. *BMJ* **2021**, *372*, n71. [\[CrossRef\]](#)
113. Prancutè, R. Web of Science (WoS) and Scopus: The Titans of Bibliographic Information in Today's Academic World. *Publications* **2021**, *9*, 12. [\[CrossRef\]](#)
114. Mongeon, P.; Paul-Hus, A. The Journal Coverage of Web of Science and Scopus: A Comparative Analysis. *Scientometrics* **2016**, *106*, 213–228. [\[CrossRef\]](#)
115. Martinez-Cantin, R. Bayesian Optimization with Adaptive Kernels for Robot Control. In Proceedings of the 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, 29 May–3 June 2017; pp. 3350–3356.
116. Hebbal, A.; Brevault, L.; Balesdent, M.; Talbi, E.G.; Melab, N. Bayesian Optimization Using Deep Gaussian Processes with Applications to Aerospace System Design. *Optim. Eng.* **2021**, *22*, 321–361. [\[CrossRef\]](#)
117. Taylor, C.J.; Felton, K.C.; Wigh, D.; Jeraal, M.I.; Grainger, R.; Chessari, G.; Johnson, C.N.; Lapkin, A.A. Accelerated Chemical Reaction Optimization Using Multi-Task Learning. *ACS Cent. Sci.* **2023**, *9*, 957–968. [\[CrossRef\]](#)
118. Suvarna, M.; Zou, T.; Chong, S.H.; Ge, Y.; Martín, A.J.; Pérez-Ramírez, J. Active Learning Streamlines Development of High Performance Catalysts for Higher Alcohol Synthesis. *Nat. Commun.* **2024**, *15*, 5844. [\[CrossRef\]](#) [\[PubMed\]](#)
119. Hoang, K.T.; Boersma, S.; Mesbah, A.; Imsland, L. Heteroscedastic Bayesian Optimisation for Active Power Control of Wind Farms. *IFAC-PapersOnLine* **2023**, *56*, 7650–7655. [\[CrossRef\]](#)
120. Buisson-Fenet, M.; Solowjow, F.; Trimpe, S. Actively Learning Gaussian Process Dynamics. In *Proceedings of the 2nd Conference on Learning for Dynamics and Control, Berkeley, CA, USA, 10–11 June 2020*; Proceedings of Machine Learning Research; PMLR: Cambridge, MA, USA, 2020; Volume 120, pp. 5–15.
121. Gafurov, A.N.; Lee, S.; Ali, U.; Irfan, M.; Kim, I.; Lee, T.M. AI-driven Digital Twin for Autonomous Web Tension Control in Roll-to-Roll Manufacturing System. *Sci. Rep.* **2025**, *15*, 24096. [\[CrossRef\]](#)
122. Brown, L.A.; Fernandes, R.; Verrelst, J.; Morris, H.; Djamai, N.; Reyes-Muñoz, P.; D.Kovács, D.; Meier, C. GROUNDED EO: Data-driven Sentinel-2 LAI and FAPAR Retrieval Using Gaussian Processes Trained with Extensive Fiducial Reference Measurements. *Remote Sens. Environ.* **2025**, *326*, 114797. [\[CrossRef\]](#)
123. Haas, M.; Onuseit, V.; Powell, J.; Zaiß, F.; Wahl, J.; Menold, T.; Hagenlocher, C.; Michalowski, A. Improving the Weld Seam Quality in Laser Welding Processes by Means of Bayesian Optimization. *Procedia CIRP* **2024**, *124*, 772–775. [\[CrossRef\]](#)
124. Karkaria, V.; Goeckner, A.; Zha, R.; Chen, J.; Zhang, J.; Zhu, Q.; Cao, J.; Gao, R.X.; Chen, W. Towards a Digital Twin Framework in Additive Manufacturing: Machine Learning and Bayesian Optimization for Time Series Process Optimization. *J. Manuf. Syst.* **2024**, *75*, 322–332. [\[CrossRef\]](#)
125. Johnson, J.E.; Jamil, I.R.; Pan, L.; Lin, G.; Xu, X. Bayesian Optimization with Gaussian-process-based Active Machine Learning for Improvement of Geometric Accuracy in Projection Multi-photon 3D Printing. *Light Sci. Appl.* **2025**, *14*, 56. [\[CrossRef\]](#)
126. Maier, M.; Zwicker, R.; Akbari, M.; Rupenyan, A.; Wegener, K. Bayesian Optimization for Autonomous Process Set-up in Turning. *CIRP J. Manuf. Sci. Technol.* **2019**, *26*, 81–87. [\[CrossRef\]](#)

127. Kavas, B.; Balta, E.C.; Tucker, M.R.; Krishnadas, R.; Rupenyan, A.; Lygeros, J.; Bambach, M. In-situ Controller Autotuning by Bayesian Optimization for Closed-loop Feedback Control of Laser Powder Bed Fusion Process. *Addit. Manuf.* **2025**, *99*, 104641. [\[CrossRef\]](#)
128. Gnanasambandam, R.; Shen, B.; Law, A.C.C.; Dou, C.; Kong, Z. Deep Gaussian Process for Enhanced Bayesian Optimization and Its Application in Additive Manufacturing. *IIE Trans.* **2025**, *57*, 423–436. [\[CrossRef\]](#)
129. Li, G.; Wang, Y.; Kar, S.; Jin, X. Bayesian Optimization with Active Constraint Learning for Advanced Manufacturing Process Design. *IIE Trans.* **2026**, *58*, 257–271. [\[CrossRef\]](#)
130. Khatamsaz, D.; Neuberger, R.; Roy, A.M.; Zadeh, S.H.; Otis, R.; Arróyave, R. A Physics Informed Bayesian Optimization Approach for Material Design: Application to NiTi Shape Memory Alloys. *npj Comput. Mater.* **2023**, *9*, 221. [\[CrossRef\]](#)
131. Lee, C.; Wang, K.; Wu, J.; Cai, W.; Yue, X. Partitioned Active Learning for Heterogeneous Systems. *J. Comput. Inf. Sci. Eng.* **2023**, *23*, 041009. [\[CrossRef\]](#)
132. Christianson, R.B.; Gramacy, R.B. Robust Expected Improvement for Bayesian Optimization. *IIE Trans.* **2024**, *56*, 1294–1306. [\[CrossRef\]](#)
133. Li, C.Y.; Rakitsch, B.; Zimmer, C. Safe Active Learning for Multi-Output Gaussian Processes. In *Proceedings of the 25th International Conference on Artificial Intelligence and Statistics (AISTATS), Virtual, 28–30 March 2022*; Proceedings of Machine Learning Research; PMLR: Cambridge, MA, USA, 2022; Volume 151, pp. 4512–4551.
134. Qian, H.M.; Wei, J.; Huang, H.Z.; Dong, Q.; Li, Y.F. Kriging-based Reliability Analysis for a Multi-output Structural System with Multiple Response Gaussian Process. *Qual. Reliab. Eng. Int.* **2023**, *39*, 1622–1638. [\[CrossRef\]](#)
135. Yu, Y.; Ma, D.; Yang, M.; Yang, X.; Guan, H. Surrogate Modeling with Non-stationary-noise Based Gaussian Process Regression and K-Fold ANN for Systems Featuring Uneven Sensitivity Distribution. *Aerosp. Sci. Technol.* **2024**, *150*, 109157. [\[CrossRef\]](#)
136. Kim, J.; Yi, S.R.; Song, J. Estimation of First-passage Probability under Stochastic Wind Excitations by Active-learning-based Heteroscedastic Gaussian Process. *Struct. Saf.* **2023**, *100*, 102268. [\[CrossRef\]](#)
137. Wang, Y.; Pan, H.; Shi, Y.; Wang, R.; Wang, P. A new active-learning estimation method for the failure probability of structural reliability based on Kriging model and simple penalty function. *Comput. Methods Appl. Mech. Eng.* **2023**, *410*, 116035. [\[CrossRef\]](#)
138. Chun, J. Active Learning-Based Kriging Model with Noise Responses and Its Application to Reliability Analysis of Structures. *Appl. Sci.* **2024**, *14*, 882. [\[CrossRef\]](#)
139. Park, C.; Waelder, R.; Kang, B.; Maruyama, B.; Hong, S.; Gramacy, R.B. Active Learning of Piecewise Gaussian Process Surrogates. *Technometrics* **2026**, *68*, 186–201. [\[CrossRef\]](#)
140. Astudillo, R.; Frazier, P. Bayesian Optimization of Function Networks. In *Proceedings of the Advances in Neural Information Processing Systems, Virtual, 6–14 December 2021*; Curran Associates, Inc.: Red Hook, NY, USA, 2021; Volume 34, pp. 14463–14475.
141. Buathong, P.; Wan, J.; Astudillo, R.; Daulton, S.; Balandat, M.; Frazier, P.I. Bayesian Optimization of Function Networks with Partial Evaluations. In *Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria, 21–27 July 2024*; Proceedings of Machine Learning Research; PMLR: Cambridge, MA, USA, 2024; Volume 235, pp. 4752–4784.
142. Lartaud, P.; Humbert, P.; Garnier, J. Solving Bayesian Inverse Problems Using Gaussian Process Regression with Goal-oriented Active Learning. *Technometrics* **2026**, *68*, 172–185. [\[CrossRef\]](#)
143. Leco, M.; McLeay, T.; Kadirkamanathan, V. A Two-step Machining and Active Learning Approach for Right-first-time Robotic Countersinking through In-process Error Compensation and Prediction of Depth of Cuts. *Robot. Comput.-Integr. Manuf.* **2022**, *77*, 102345. [\[CrossRef\]](#)
144. Schreiter, J.; Nguyen-Tuong, D.; Eberts, M.; Bischoff, B.; Markert, H.; Toussaint, M. Safe Exploration for Active Learning with Gaussian Processes. In *Machine Learning and Knowledge Discovery in Databases*; Springer International Publishing: Cham, Switzerland, 2015; Volume 9286, pp. 133–149.
145. Park, S.M.; Lee, T.; Lee, J.H.; Kang, J.S.; Kwon, M.S. Gaussian Process Regression-based Bayesian Optimization of the Insulation-coating Process for Fe–Si Alloy sheets. *J. Mater. Res. Technol.* **2023**, *22*, 3294–3301. [\[CrossRef\]](#)
146. Prentzas, I.; Fragiadakis, M. Quantified Active Learning Kriging Model for Structural Reliability Analysis. *Probabilistic Eng. Mech.* **2024**, *78*, 103699. [\[CrossRef\]](#)
147. Patel, R.A.; Kesharwani, S.S.; Ibrahim, F. Active Learning and Gaussian Processes for the Development of Dissolution Models: An AI-based Data-efficient Approach. *J. Control. Release* **2025**, *379*, 316–326. [\[CrossRef\]](#)
148. Ghosh, S.; Berkenkamp, F.; Ranade, G.; Qadeer, S.; Kapoor, A. Verifying Controllers Against Adversarial Examples with Bayesian Optimization. In *Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, Australia, 21–25 May 2018*; pp. 7306–7313.
149. Maddox, W.J.; Balandat, M.; Wilson, A.G.; Bakshy, E. Bayesian Optimization with High-Dimensional Outputs. In *Proceedings of the Advances in Neural Information Processing Systems, Virtual, 6–14 December 2021*; Curran Associates, Inc.: Red Hook, NY, USA, 2021; Volume 34, pp. 19274–19287.
150. Lu, M.; Li, H.; Hong, L. An Adaptive Kriging Reliability Analysis Method Based on Novel Condition Likelihood Function. *J. Mech. Sci. Technol.* **2022**, *36*, 3911–3922. [\[CrossRef\]](#)

151. Kontoudis, G.P.; Otte, M. Adaptive Exploration-Exploitation Active Learning of Gaussian Processes. In Proceedings of the 2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Detroit, MI, USA, 1–5 October 2023; pp. 9448–9455.
152. Chitturi, S.R.; Ramdas, A.; Wu, Y.; Rohr, B.; Ermon, S.; Dionne, J.; Da Jornada, F.H.; Dunne, M.; Tassone, C.; Neiswanger, W.; et al. Targeted Materials Discovery Using Bayesian Algorithm Execution. *npj Comput. Mater.* **2024**, *10*, 156. [\[CrossRef\]](#)
153. Denkena, B.; Wichmanna, M.; Rokicki, M.; Stürenburg, L. Active Learning for the Prediction of Shape Errors in Milling. *Procedia CIRP* **2024**, *126*, 324–329. [\[CrossRef\]](#)
154. Gardner, J.; Pleiss, G.; Weinberger, K.Q.; Bindel, D.; Wilson, A.G. GPyTorch: Blackbox Matrix-Matrix Gaussian Process Inference with GPU Acceleration. *Adv. Neural Inf. Process. Syst.* **2018**, *31*, 7576–7586.
155. Bingham, E.; Chen, J.P.; Jankowiak, M.; Obermeyer, F.; Pradhan, N.; Karaletsos, T.; Singh, R.; Szerlip, P.; Horsfall, P.; Goodman, N.D. Pyro: Deep Universal Probabilistic Programming. *J. Mach. Learn. Res.* **2019**, *20*, 973–978.
156. Huang, H.; Xu, T.; Xi, Y.; Chow, E. HiGP: A high-performance Python package for Gaussian Processes. *J. Open Source Softw.* **2026**, *11*, 8621. [\[CrossRef\]](#)
157. Matthews, A.G.d.G.; van der Wilk, M.; Nickson, T.; Fujii, K.; Boukouvalas, A.; Leon-Villagrà, P.; Ghahramani, Z.; Hensman, J. GPflow: A Gaussian Process Library using TensorFlow. *J. Mach. Learn. Res.* **2017**, *18*, 1–6.
158. Dutordoir, V.; Salimbeni, H.; Hambro, E.; McLeod, J.; Leibfried, F.; Artemev, A.; van der Wilk, M.; Hensman, J.; Deisenroth, M.P.; John, S.T. GPflux: A Library for Deep Gaussian Processes. *arXiv* **2021**, arXiv:2104.05674. [\[CrossRef\]](#)
159. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Müller, A.; Nothman, J.; Louppe, G.; et al. Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.
160. Saves, P.; Lafage, R.; Bartoli, N.; Diouane, Y.; Bussemaker, J.; Lefebvre, T.; Hwang, J.T.; Morlier, J.; Martins, J.R. SMT 2.0: A Surrogate Modeling Toolbox with a Focus on Hierarchical and Mixed Variables Gaussian Processes. *Adv. Eng. Softw.* **2024**, *188*, 103571. [\[CrossRef\]](#)
161. de Wolff, T.; Cuevas, A.; Tobar, F. MOGPTK: The Multi-output Gaussian Process Toolkit. *Neurocomputing* **2021**, *424*, 49–53. [\[CrossRef\]](#)
162. O’Gara, D.; Binois, M.; Garnett, R.; Hammond, R.A. hetGPy: Heteroskedastic Gaussian Process Modeling in Python. *J. Open Source Softw.* **2025**, *10*, 7518. [\[CrossRef\]](#)
163. Salvatier, J.; Wiecki, T.V.; Fonnesbeck, C. Probabilistic Programming in Python Using PyMC3. *PeerJ Comput. Sci.* **2016**, *2*, e55. [\[CrossRef\]](#)
164. Olson, M.; Santorella, E.; Tiao, L.C.; Cakmak, S.; Eriksson, D.; Garrard, M.; Daulton, S.; Balandat, M.; Bakshy, E.; Kashtelyan, E.; et al. Ax: A Platform for Adaptive Experimentation. In Proceedings of the Fourth International Conference on Automated Machine Learning, New York City, NY, USA, 8–11 September 2025; Proceedings of Machine Learning Research; PMLR: Cambridge, MA, USA, 2025; Volume 293.
165. Moss, H.; Picheny, V.; Stojic, H.; Ober, S.W.; Artemev, A.; Paleyes, A.; Vakili, S.; Markou, S.; Qing, J.; Loka, N.R.B.S.; et al. Trieste: Efficiently Exploring The Depths of Black-box Functions with TensorFlow. In Proceedings of the NeurIPS 2024 Workshop on Bayesian Decision-making and Uncertainty, Vancouver, BC, Canada, 14 December 2024.
166. Paleyes, A.; Mahserici, M.; Lawrence, N. Emukit: A Python Toolkit for Decision Making under Uncertainty. In Proceedings of the 22nd Python in Science Conference, SciPy, Austin, TX, USA, 10–16 July 2023; pp. 68–75.
167. Akiba, T.; Sano, S.; Yanase, T.; Ohta, T.; Koyama, M. Optuna. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Anchorage, AK, USA, 4–8 August 2019; pp. 2623–2631.
168. Danka, T.; Horvath, P. modAL: A Modular Active Learning Framework for Python. *arXiv* **2018**, arXiv:1805.00979. [\[CrossRef\]](#)
169. Kandasamy, K.; Vysyaraju, K.R.; Neiswanger, W.; Paria, B.; Collins, C.R.; Schneider, J.; Poczos, B.; Xing, E.P. Tuning Hyperparameters without Grad Students: Scalable and Robust Bayesian Optimisation with Dragonfly. *J. Mach. Learn. Res.* **2020**, *21*, 1–27.
170. Gramacy, R.B. tgp: An R Package for Bayesian Nonstationary, Semiparametric Nonlinear Regression and Design by Treed Gaussian Process Models. *J. Stat. Softw.* **2007**, *19*, 1–46. [\[CrossRef\]](#)
171. Gramacy, R.B. laGP: Large-Scale Spatial Modeling via Local Approximate Gaussian Processes in R. *J. Stat. Softw.* **2016**, *72*, 1–46. [\[CrossRef\]](#)
172. Binois, M.; Gramacy, R.B. hetGP: Heteroskedastic Gaussian Process Modeling and Sequential Design in R. *J. Stat. Softw.* **2021**, *98*, 1–44. [\[CrossRef\]](#)
173. Roustant, O.; Ginsbourger, D.; Deville, Y. DiceKriging, DiceOptim: Two R Packages for the Analysis of Computer Experiments by Kriging-Based Metamodeling and Optimization. *J. Stat. Softw.* **2012**, *51*, 1–55. [\[CrossRef\]](#)
174. Binois, M.; Picheny, V. GPareto: An R Package for Gaussian-Process-Based Multi-Objective Optimization and Analysis. *J. Stat. Softw.* **2019**, *89*, 1–30. [\[CrossRef\]](#)

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.